4

The Non-associative Lambek Calculus

Summary. In this chapter we will look at NL, the non-associative Lambek calculus, which was introduced by Lambek a few years after the Syntactic Calculus, L. The Gentzen-style presentation of the Lambek calculus uses a list of formulae as antecedents, whereas NL uses binary branching trees instead.

We will start this chapter with a brief introduction of the sequent calculus for NL, then illustrate why non-associativity is sometime desirable by presenting some ungrammatical sentences which — though derivable in L — are underivable in NL.

We will then revisit some of the results of L from the perspective of the non-associative calculus NL: we will reprove cut elimination for NL and present a natural deduction version of the calculus.

In addition, we will give a new type of model for Lambek calculi in the form of Kripke models, for which we prove soundness and completeness. We will also show how we can add an explicit rule of associativity to NL to obtain an alternative formulation of L and see how this corresponds to a constraint on the Kripke models.

Perhaps surprisingly, dropping the structural rule of associativity makes a big computational difference: whereas the Lambek calculus has been shown to be NP complete by Pentus, the non-associative Lambek calculus has a polynomial time decision algorithm. We will finish our investigation of NL by presenting the polynomial time algorithms of Aarts & Trautwein and of de Groote.

4.1 Introduction

The non-associative Lambek calculus NL is obtained from the Lambek calculus by dropping the (implicit) rule of associativity. Instead of using *lists* of formulae as antecedents, like we did for L, NL uses binary branching *trees* of formulae.

Since in many linguistic frameworks, the basic units of linguistic description are considered to be trees — notably in the Chomskyan tradition (Chomsky, 1982, 1995; Stabler, 1997) but also in several alternative frameworks such as tree adjoining grammars (Joshi and Schabes, 1997) or HPSG (Pollard and Sag, 1994), where

R. Moot and C. Retoré: The Logic of Categorial Grammars 2012, LNCS 6850, pp. 101–147, 2012.

the *daughters* feature encodes the tree information — it makes sense to investigate the non-associative Lambek calculus and see if it offers any advantages over the Lambek calculus.

4.2 Proof Theory

In order not to overburden the notation, we will use the same symbols for the connectives of NL as for L. Unless otherwise indicated in the text, the formulae we will talk about in this chapter will be the formulae of NL.

$$\mathsf{Lp} \quad ::= \quad \mathsf{P} \quad | \quad (\mathsf{Lp} \backslash \mathsf{Lp}) \quad | \quad (\mathsf{Lp} / \mathsf{Lp}) \quad | \quad (\mathsf{Lp} \bullet \mathsf{Lp})$$

In L, the antecedent of a sequent was a (non-empty) list of formulae, which had the convenience of making the rule of associativity implicit. For NL, we want to drop this implicit rule of associativity and this means using a binary-branching *tree*, with formulae as its leaves, as antecedents. We will call these binary-branching trees *antecedent terms*.

Definition 4.1. *The antecedent terms $\mathscr{A}$ are defined as follows.*

$$\mathscr{A} \quad ::= \quad \mathsf{Lp} \quad | \quad (\mathscr{A}, \mathscr{A})$$

So for example $(np, np\backslash S)$ and $(((a, b/c), d), e)$ are antecedent terms (if the atomic formulae include np and S in the first case and a, b, c, d and e in the second).

In the following Γ, Δ will denote antecedent terms.

We define some basic functions which transform the tree-structured antecedent terms to lists (simply by removing the brackets) and to multisets (keeping only the formulae and the number of occurrences of each formula, but forgetting the order).

Definition 4.2. *Let Γ be an antecedent term, the* yield *of Γ is a list which is obtained as follows.*

$$\begin{aligned} yield(F) &= F \qquad \text{if } F \text{ is a formula} \\ yield(\Gamma, \Delta) &= yield(\Gamma), yield(\Delta) \end{aligned}$$

The comma is deliberately overloaded here, so that *yield* transforms an NL antecedent term into a valid (and non-empty) L list.

Definition 4.3. *Let Γ be an antecedent term, the* multiset of formulae *of Γ is defined by the function formulae(Γ) as follows.*

$$\begin{aligned} formulae(F) &= \{F\} \qquad \text{if } F \text{ is a formula} \\ formulae(\Gamma, \Delta) &= formulae(\Gamma) \cup formulae(\Delta) \end{aligned}$$

where $\cup$ is the multiset union operation.

So $formulae((np, np\backslash S)) = \{np, np\backslash S\}$ and $formulae(((a,b),(c,(b,a)))) = \{a,a,b,b,c\}$.

For the definition of the sequent rules, it is necessary to refer to *contexts*, which are defined as follows.

Definition 4.4. *A* context *is defined as follows.*

$$\mathscr{C} \quad ::= \quad [] \quad | \quad (\mathscr{C},\mathscr{A}) \quad | \quad (\mathscr{A},\mathscr{C})$$

where $\mathscr{A}$ is an antecedent term according to Definition 4.1.

A context is an antecedent term with a single occurrence of a 'hole' denoted by '[]'; seen this way, the inductive definition defines a path to the hole, with the three cases corresponding to 'here', 'on the left branch' and 'on the right branch' respectively.

We will write $\Gamma[]$, $\Delta[]$ to denote contexts.

Definition 4.5. *The* substitution *of an antecedent term Δ in a context $\Gamma[]$, $subst(\Gamma[], \Delta)$ (which we will normally write simply as $\Gamma[\Delta]$) is defined as follows.*

$$\begin{aligned} subst([],\Delta) &= \Delta \\ subst((\Gamma,\Gamma'[]),\Delta) &= (\Gamma, subst(\Gamma'[],\Delta)) \\ subst((\Gamma[],\Gamma'),\Delta) &= (subst(\Gamma[],\Delta),\Gamma') \end{aligned}$$

Note that the substitution of an antecedent term into a context produces a valid antecedent term. We can define the substitution of a context $\Delta[]$ in a context $\Gamma[]$ analogously to Definition 4.5 above, which gives a context $\Gamma[\Delta[]]$ after substitution.

4.2.1 Sequent Calculus

We now have everything in place for giving the sequent calculus formulation of NL, which is shown in Figure 4.1.

$$\frac{\Gamma[B] \vdash C \quad \Delta \vdash A}{\Gamma[(\Delta, A \backslash B)] \vdash C} \backslash_h \qquad \frac{(A,\Gamma) \vdash C}{\Gamma \vdash A \backslash C} \backslash_i$$

$$\frac{\Gamma[B] \vdash C \quad \Delta \vdash A}{\Gamma[(B / A, \Delta)] \vdash C} /_h \qquad \frac{(\Gamma,A) \vdash C}{\Gamma \vdash C / A} /_i$$

$$\frac{\Gamma[(A,B)] \vdash C}{\Gamma[A \bullet B] \vdash C} \bullet_h \qquad \frac{\Delta \vdash A \quad \Gamma \vdash B}{(\Delta,\Gamma) \vdash A \bullet B} \bullet_i$$

$$\frac{\Gamma \vdash A \quad \Delta[A] \vdash B}{\Delta[\Gamma] \vdash B} cut \qquad \frac{}{A \vdash A} axiom$$

Fig. 4.1. Sequent calculus rule for NL, the non-associative Lambek calculus

NL does not allow empty antecedent derivations, which we have identified as undesirable in Section 2.5: the antecedent term Γ in the $\backslash_i$ and the $/_i$ rules is non-empty according to Definition 4.1.

When talking about the derivability of a sequent in the non-associative Lambek calculus, we can ask two different questions:

1. Given a list of formulae L and a goal formula C, for which antecedent terms Γ such that $\mathit{yield}(\Gamma) = L$, is $\Gamma \vdash C$ derivable?
2. Given an antecedent term Γ and a goal formula C, is $\Gamma \vdash C$ derivable?

That is, when we look at sequent proof search, do we consider the structure of the antecedent term to be part of the *output* of the proof search algorithm (as in option 1 above) or as part of its *input* (as in option 2). In other words, do we compute the brackets of the antecedent terms or just verify them? In what follows, unless otherwise noted — for example in Section 4.6 — when talking about parsing or proof search, we will consider the input of the parsing or proof search algorithm to be a list of formulae (or a list of *words*, when using a lexicon which assigns sets of formulae to these words). When there is a need to emphasize that the structure of the antecedent term of a sequent is unknown, we will write the sequent as $A_1, \dots, A_n \vdash C$, just like we did for Lambek calculus sequents.

If we define the yield of an antecedent term of the form $\Gamma[\Delta]$ (for some Δ) to be $\Gamma, \mathit{yield}(\Delta), \Gamma'$ (with Γ' an unused antecedent term variable), then it is easy to verify that the following proposition holds.

Proposition 4.6. *If R is an NL sequent rule and $\Gamma_1, \dots, \Gamma_n$ are the antecedent terms mentioned in R then replacing each Γ_i by $\mathit{yield}(\Gamma_i)$ will give an L sequent rule.*

Corollary 4.7. *If $\Gamma \vdash A$ is derivable in NL then $\mathit{yield}(\Gamma) \vdash A$ is derivable in L.*

The inverse does not hold however: some characteristic theorems of L are underivable in NL.

Example 4.8. An example is the transitivity of /, as shown by the following failed derivation.

$$\dfrac{\dfrac{(A,C) \vdash A \quad B/C \vdash B}{((A/B, B/C), C) \vdash A}\,/_h}{(A/B, B/C) \vdash A/C}\,/_i$$

Note how the parentheses prevent application of the $/_h$ rule to B/C and C, as they are not sisters in the tree. Showing a failed proof attempt is not that same as showing underivability of a sequent, however! We need to show that *all* proof attempts fail. Exercise 4.1 asks you to verify by means of an exhaustive proof search that the sequent of this example has no proof in NL.

Showing non-derivability by exhaustive enumeration of proof attempts can be tedious and error-prone. We will see methods requiring less bookkeeping to show underivability in Section 4.6 and in Chapter 7. In Section 4.5.2 we will see another method to show a sequent is not derivable: the construction of a countermodel.

The count check of Proposition 2.6 is useful but incomplete. For example $(A / B, B / C) \vdash A / C$ satisfies the count check, but — as we have seen above — it is underivable nonetheless.

4.2.2 Arguments against Associativity

As an illustration of why associativity is sometimes undesirable, look at the following lexicon (after Lambek (1961)).

Word	Type(s)
the	np / n
Hulk	n
is	$(np \backslash s) / (n / n)$
incredible	n / n
green	n / n

Using this lexicon, both NL and L allow us the derive the following phrases.

(4.1) The Hulk is green.
(4.2) The Hulk is incredible.

However, L allows us the derive the following ungrammatical phrase as well.

(4.3) $*$ The Hulk is green incredible.

Exercise 4.3 at the end of this chapter asks you to show the non-derivability of

$$n / n, n / n \vdash n / n$$

in NL and to show that only the valid example sentences above are derivable in NL, whereas the invalid sentence 4.3 is derivable in L.

There is another type of example to show that associativity can lead to some very strange sentences. In order to present this argument, we need some introduction to the use of so-called *polymorphic* types, which can be used to conjoin expression which are assigned different formulae, as demonstrated by the examples below (Exercise 1.4.4 gives several other examples).

(4.4) Bill left the party and returned home.
(4.5) Bill gave flowers to Mary and a toy to the children.

Example 4.4 above shows that "left the party" and "returned home", both expressions of type $np \backslash S$, can be conjoined. This means that "and" can be assigned the formula $((np \backslash S) \backslash (np \backslash S)) / (np \backslash S)$, as well as many other instances of the general scheme $(X \backslash X) / X$.[1]

Example 4.5 shows that the items conjoined can be complex expressions such as "flowers to Mary" and "a toy to the children", which in the context of a formula assignment of $((np \backslash S) / pp) / np$ to "gave" makes "flowers to Mary" an expression of type $np \bullet pp$.

Now, with these examples in mind, look at the following sentence.

(4.6) *The mother of and Bill thought John arrived.

This sentence, as Paul Dekker was the first to notice, is not only clearly ungrammatical, but also — though this may seem surprising (and even shocking!) at first glance — derivable in L: there is an instantiation of the polymorphic type scheme $(X \backslash X) / X$ for "and" which makes the sentence derivable in the Lambek calculus: both "the mother of" and "Bill thought" can be shown to be of type $(S / (np \backslash S)) / np$ using the following lexicon, which is without surprises,

Word	Type(s)
Bill	np
John	np
the	np / n
mother	n / pp
of	pp / np
thought	$((np \backslash S) / S)$
arrived	$np \backslash S$

as shown by the following two natural deduction proofs in the associative Lambek calculus (using the Prawitz-style rules of Section 2.2.1 and the rule Lex to indicate the conclusion of the rule is a lexical entry for the word which serves as its premise)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{the}}{np / n}\,\text{Lex} \quad \dfrac{\dfrac{\text{mother}}{n / pp}\,\text{Lex} \quad \dfrac{\dfrac{\text{of}}{pp / np}\,\text{Lex} \quad [np]^1}{pp}\,/e}{n}\,/e}{np}\,/e \quad [np \backslash S]^2}{S}\,\backslash e}{S / (np \backslash S)}\,/i_2}{(S / (np \backslash S)) / np}\,/i_1$$

[1] This scheme corresponds to a quantification over formulas and it would be more correct to write $\forall X.(X \backslash X) / X$. Though a natural extension of L, the resulting calculus is undecidable, so this extension is not as innocent as it may appear (see Emms, 1993, 1995).

$$\dfrac{\dfrac{\text{Bill}}{np}\,\text{Lex} \qquad \dfrac{\dfrac{\text{thought}}{(np\backslash S)/S}\,\text{Lex} \qquad \dfrac{[np]^1 \quad [np\backslash S]^2}{S}\,\backslash e}{np\backslash S}\,/e}{\dfrac{S}{\dfrac{S/(np\backslash S)}{(S/(np\backslash S))/np}\,/i_1}\,/i_2}\,\backslash e$$

Neither proof can be transformed into a valid NL proof.

In some cases, however, the absence of associativity excludes some *grammatical* sentences as well. For example, the elegant treatment of peripheral extraction, as exemplified by the Italian sentence from Example 2.2 and the sentences of Exercise 2.7, is invalid in NL. Exercises 4.5 and 4.6 at the end of this chapter ask you to prove this. Similar remarks can be made about the quantifier scope ambiguities of the previous chapter (see Example 3.5). Exercise 4.9 asks you to verify that the type for object quantifiers in L cannot be used for NL.

So, summing up, we have seen that in some cases — in combination with coordination of in combination with the most natural type assignments for the verb "to be" and adjectives — it is desirable to restrict associativity, whereas in other cases — in the case of quantifier scope and of peripheral extraction — the easiest solution would be to permit associativity. In the next chapter, we will see how to combine an associative and a non-associative logic into one system. For the rest of this chapter, we will study the non-associative calculus (though, as a preamble to this, we will show in Section 4.3 how to add structural rules to NL and recover L).

4.2.3 Cut Elimination for the NL Sequent Calculus

In order to verify that cut elimination is valid for NL, following (Lambek, 1961; Kandulski, 1988), we revisit the different cases of the proof for L and verify that the bracketing is respected. This is just a simple exercise, but something we need to do to verify our logic is formulated correctly.

Remember that we are in the following general case for a cut rule of depth r and degree d and that we are looking at a cut rule of smallest depth in the proof.

$$\dfrac{\dfrac{\vdots\,\gamma}{\Gamma\vdash D}\,R^a \qquad \dfrac{\vdots\,\delta}{\Delta[D]\vdash C}\,R^f}{\Delta[\Gamma]\vdash C}\,\mathit{cut}\ d$$

We look at rule R^f and R^a. Since this is the rule with the smallest depth in the proof, there are no other cut rules in either γ or δ. We look at the other cases, which are the same as before: 1) at least one of the rules is an axiom, 2) R^a is not the rule which creates the cut formula, 3) R^f is not the rule which creates the cut formula or 4) both R^a and R^f create the cut formula.

1. If at least one of the rules is an axiom, we can remove the cut as follows.

$$\frac{D \vdash D \qquad \begin{array}{c}\vdots\,\delta\\ \Gamma[D]\vdash C\end{array}}{\Gamma[D]\vdash C}\,cut \qquad \text{reduces to} \qquad \begin{array}{c}\vdots\,\delta\\ \Gamma[D]\vdash C\end{array}$$

2. If R^a does not create the cut formula D, then we move the rule up past R^a. We know R^a must have been one of $\backslash_h$, $/_h$, $\bullet_h$, since an introduction rule would have necessarily introduced the cut formula. The table below lists the different cases (the implications are symmetric, so only $\backslash$ is shown).

2 $\;R^a$ **does not create** D**, the cut formula**		
R^a	Before reduction	After reduction
$\bullet_h$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\gamma\\ \Delta[(A,B)]\vdash D\end{array}}{\Delta[A\bullet B]\vdash D}\,\bullet_h \quad \begin{array}{c}\vdots\,\delta\\ \Gamma[D]\vdash C\end{array}}{\Gamma[\Delta[A\bullet B]]\vdash C}\,cut\ d$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\gamma\\ \Delta[(A,B)]\vdash D\end{array} \quad \begin{array}{c}\vdots\,\delta\\ \Gamma[D]\vdash C\end{array}}{\Gamma[\Delta[(A,B)]]\vdash C}\,cut\ d}{\Gamma[\Delta[A\bullet B]]\vdash C}\,\bullet_h$
$\backslash_h$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\delta\\ \Delta'[B]\vdash D\end{array} \quad \begin{array}{c}\vdots\,\delta'\\ \Delta\vdash A\end{array}}{\Delta'[(\Delta,A\backslash B)]\vdash D}\,\backslash_h \quad \begin{array}{c}\vdots\,\gamma\\ \Gamma[D]\vdash C\end{array}}{\Gamma[\Delta'[(\Delta,A\backslash B)]]\vdash C}\,cut\ d$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\delta\\ \Delta'[B]\vdash D\end{array} \quad \begin{array}{c}\vdots\,\gamma\\ \Gamma[D]\vdash C\end{array}}{\Gamma[\Delta'[B]]\vdash C}\,cut\ d \quad \begin{array}{c}\vdots\,\delta'\\ \Delta\vdash A\end{array}}{\Gamma[\Delta'[(\Delta,A\backslash B)]]\vdash C}\,\backslash_h$

3. If R^f does not create the cut formula, then we move the rule up past R^f. There are rather many cases to consider.

   For the left rules $\backslash_h$, $/_h$, $\bullet_h$, since the cut formula is not the main formula of the rule, the cut formula is a formula in $\Gamma[]$ (in the case of $\backslash_h$ and $/_h$ it can be a formula of Θ as well, hence the alternative case in the cut elimination below), which is already a context. Instead of introducing a new type of context with two distinguished formulae — D and the main formula of the rule — we simply indicate that D is a formula in $\Gamma[]$ and write $\Gamma^{\{D:=\Delta\}}[]$ for the context $\Gamma[]$ where D has been replaced by Δ in the reductions for $\backslash_h$ and $\bullet_h$ (the reduction for $/_h$ is symmetric to the reduction for $\backslash_h$ and has been omitted).

3 R^f **does not create** D**, the cut formula**		
R^f	Before reduction	After reduction
$\bullet_h$	$\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \dfrac{\overset{\vdots\,\gamma}{\Gamma[(A,B)] \vdash C}}{\Gamma[A \bullet B] \vdash C}\bullet_h}{\Gamma^{\{D:=\Delta\}}[A \bullet B] \vdash C}\,cut\ d$	$\dfrac{\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \overset{\vdots\,\gamma}{\Gamma[(A,B)] \vdash C}}{\Gamma^{\{D:=\Delta\}}[(A,B)] \vdash C}\,cut\ d}{\Gamma^{\{D:=\Delta\}}[A \bullet B] \vdash C}\bullet_h$
$\backslash_h$	$\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \dfrac{\overset{\vdots\,\gamma}{\Gamma[B] \vdash C} \quad \overset{\vdots\,\theta}{\Theta \vdash A}}{\Gamma[(\Theta, A \backslash B)] \vdash C}\backslash_h}{\Gamma^{\{D:=\Delta\}}[(\Theta, A \backslash B)] \vdash C}\,cut\ d$	$\dfrac{\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \overset{\vdots\,\gamma}{\Gamma[B] \vdash C}}{\Gamma^{\{D:=\Delta\}}[B] \vdash C}\,cut\ d \quad \overset{\vdots\,\theta}{\Theta \vdash A}}{\Gamma^{\{D:=\Delta\}}[(\Theta, A \backslash B)] \vdash C}\backslash_h$
$\backslash_h$	$\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \dfrac{\overset{\vdots\,\gamma}{\Gamma[B] \vdash C} \quad \overset{\vdots\,\theta}{\Theta[D] \vdash A}}{\Gamma[(\Theta[D], A \backslash B)] \vdash C}\backslash_h}{\Gamma[(\Theta[\Delta], A \backslash B)] \vdash C}\,cut\ d$	$\dfrac{\overset{\vdots\,\gamma}{\Gamma[B] \vdash C} \quad \dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \overset{\vdots\,\theta}{\Theta[D] \vdash A}}{\Theta[\Delta] \vdash A}\,cut\ d}{\Gamma[(\Theta[\Delta], A \backslash B)] \vdash C}\backslash_h$
$\bullet_i$	$\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \dfrac{\overset{\vdots\,\gamma}{\Gamma[D] \vdash A} \quad \overset{\vdots\,\theta}{\Theta \vdash B}}{(\Gamma[D], \Theta) \vdash A \bullet B}\bullet_i}{(\Gamma[\Delta], \Theta) \vdash A \bullet B}\,cut\ d$	$\dfrac{\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \overset{\vdots\,\gamma}{\Gamma[D] \vdash A}}{\Gamma[\Delta] \vdash A}\,cut\ d \quad \overset{\vdots\,\theta}{\Theta \vdash B}}{(\Gamma[\Delta], \Theta) \vdash A \bullet B}\bullet_i$
$\bullet_i$	$\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \dfrac{\overset{\vdots\,\gamma}{\Gamma \vdash A} \quad \overset{\vdots\,\theta}{\Theta[D] \vdash B}}{(\Gamma, \Theta[D]) \vdash A \bullet B}\bullet_i}{(\Gamma, \Theta[\Delta]) \vdash A \bullet B}\,cut\ d$	$\dfrac{\overset{\vdots\,\gamma}{\Gamma \vdash A} \quad \dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \overset{\vdots\,\theta}{\Theta[D] \vdash B}}{\Theta[\Delta] \vdash B}\,cut\ d}{(\Gamma, \Theta[\Delta]) \vdash A \bullet B}\bullet_i$
$\backslash_i$	$\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \dfrac{\overset{\vdots\,\gamma}{(A, \Gamma[D]) \vdash B}}{\Gamma[D] \vdash A \backslash B}\backslash_i}{\Gamma[\Delta] \vdash A \backslash B}\,cut\ d$	$\dfrac{\dfrac{\overset{\vdots\,\delta}{\Delta \vdash D} \quad \overset{\vdots\,\gamma}{(A, \Gamma[D]) \vdash B}}{(A, \Gamma[\Delta]) \vdash B}\,cut\ d}{\Gamma[\Delta] \vdash A \backslash B}\backslash_i$

4. Finally, in the crucial case, both rules introduce the cut formula d and we replace it by two cuts of lesser degree.

4 Both R^a and R^f create the cut-formula		
	Before reduction	After reduction
$\bullet$	$\dfrac{\dfrac{\overset{\vdots\,\delta}{\Delta\vdash A}\quad\overset{\vdots\,\theta}{\Theta\vdash B}}{(\Delta,\Theta)\vdash A\bullet B}\bullet_i\quad\dfrac{\overset{\vdots\,\gamma}{\Gamma[(A,B)]\vdash C}}{\Gamma[(A\bullet B)]\vdash C}\bullet_h}{\Gamma[(\Delta,\Theta)]\vdash C}\,cut\ d$	$\dfrac{\overset{\vdots\,\delta}{\Delta\vdash A}\quad\dfrac{\overset{\vdots\,\theta}{\Theta\vdash B}\quad\overset{\vdots\,\gamma}{\Gamma[(A,B)]\vdash C}}{\Gamma[(A,\Theta)]\vdash C}\,cut<d}{\Gamma[(\Delta,\Theta)]\vdash C}\,cut<d$
$\backslash$	$\dfrac{\dfrac{\overset{\vdots\,\delta}{(A,\Delta)\vdash B}}{\Delta\vdash A\backslash B}\backslash_i\quad\dfrac{\overset{\vdots\,\gamma}{\Gamma[B]\vdash C}\quad\overset{\vdots\,\theta}{\Theta\vdash A}}{\Gamma[(\Theta,A\backslash B)]\vdash C}\backslash_h}{\Gamma[(\Theta,\Delta)]\vdash C}\,cut\ d$	$\dfrac{\dfrac{\overset{\vdots\,\theta}{\Theta\vdash A}\quad\overset{\vdots\,\delta}{(A,\Delta)\vdash B}}{(\Theta,\Delta)\vdash B}\,cut<d\quad\overset{\vdots\,\gamma}{\Gamma[B]\vdash C}}{\Gamma[(\Theta,\Delta)]\vdash C}\,cut<d$

4.2.4 Natural Deduction

Like L, NL permits a natural deduction formulation. However, given that for NL it no longer suffices to demand that the hypothesis which is withdrawn by the introduction rules is the leftmost or rightmost hypothesis, the more explicit Gentzen

$$\dfrac{\Gamma\vdash A\quad\Delta\vdash A\backslash B}{(\Gamma,\Delta)\vdash B}\backslash_e\qquad\dfrac{(A,\Gamma)\vdash C}{\Gamma\vdash A\backslash C}\backslash_i$$

$$\dfrac{\Delta\vdash B/A\quad\Gamma\vdash A}{(\Delta,\Gamma)\vdash B}/_e\qquad\dfrac{(\Gamma,A)\vdash C}{\Gamma\vdash C/A}/_i$$

$$\dfrac{\Delta\vdash A\bullet B\quad\Gamma[(A,B)]\vdash C}{\Gamma[\Delta]\vdash C}\bullet_e\qquad\dfrac{\Delta\vdash A\quad\Gamma\vdash B}{(\Delta,\Gamma)\vdash A\bullet B}\bullet_i$$

$$\dfrac{}{A\vdash A}\,axiom$$

Fig. 4.2. Natural deduction rules for NL

style formulation introduced in Section 2.2.2 is preferred. Figure 4.2 shows the natural deduction rules for NL.

Again, with the exception of the parentheses in the current rules, the natural deduction rules for NL are the same as those of L.

4.3 Structural Rules

From NL we can recover L simply by adding the two structural rules of associativity, shown in Figure 4.3 on the left and middle, to the logic. By adding commutativity as well, as shown below to the right, two new logics become available: adding just commutativity to NL gives us the non-associative Lambek calculus with permutation NLP, whereas adding both associativity and commutativity gives the Lambek-van Benthem calculus LP.

$$\frac{\Gamma[(\Delta_1,(\Delta_2,\Delta_3))] \vdash C}{\Gamma[((\Delta_1,\Delta_2),\Delta_3)] \vdash C}\,ass1 \quad \frac{\Gamma[((\Delta_1,\Delta_2),\Delta_3)] \vdash C}{\Gamma[(\Delta_1,(\Delta_2,\Delta_3))] \vdash C}\,ass2 \quad \frac{\Gamma[(\Delta_2,\Delta_1)] \vdash C}{\Gamma[(\Delta_1,\Delta_2)] \vdash C}\,com$$

Fig. 4.3. The structural rules of associativity and commutativity

Figure 4.4 lists the four possible combinations of the structural rules and the corresponding logics.

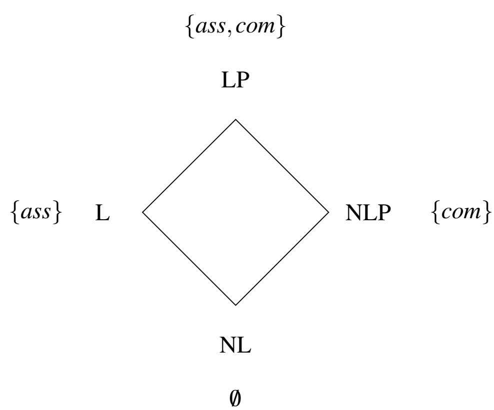


Fig. 4.4. The four logics NL, L, NLP and LP

We've already talked about the differences between NL and L and the benefits of both systems. A consequence of adding commutativity to our logic is that we can no longer distinguish between the two implications. The natural deduction proof below shows one direction, the other direction is symmetric.

$$\dfrac{\dfrac{\dfrac{\overline{A / B \vdash A / B}\ \mathit{axiom} \qquad \overline{B \vdash B}\ \mathit{axiom}}{(A / B, B) \vdash A}\ /_e}{(B, A / B) \vdash A}\ \mathit{com}}{A / B \vdash B \backslash A}\ \backslash_i$$

An example of where this property would be useful is for the treatment of English adverbs: some adverbs like ‘completely’ and ‘carefully’ can appear both before and after the verb phrase. The following two pairs of sentences, for example, should all be derivable.

(4.7) Loren carefully read Neuromancer.
(4.8) Loren read Neuromancer carefully.
(4.9) Stewart completely destroyed his credibility.
(4.10) Stewart destroyed his credibility completely.

The following lexicon allows us to derive example sentences 4.8 and 4.10 in NL. It assigns the formula $(np \backslash S) \backslash (np \backslash S)$ to the adverbs, which allows them to appear to the right of a verb phrase $np \backslash S$. We can add additional lexical entries to allow us to derive sentences 4.7 and 4.9 as well: the formula $(np \backslash S) / (np \backslash S)$ will do exactly that. So there is a trade-off to be made: do we add additional lexical entries, or do we try to generalize by adding structural rules? There is no easy answer to this question: both a small lexicon and a small set of structural rules are desirable.

Word	Type(s)
Loren	np
Stewart	np
Neuromancer	np
credibility	n
his	np / n
read	$(np \backslash S) / np$
destroyed	$(np \backslash S) / np$
carefully	$(np \backslash S) \backslash (np \backslash S)$
completely	$(np \backslash S) \backslash (np \backslash S)$

In this case, using NLP to model the behavior of adverbs and reduce the size of the lexicon has a serious drawback: it also permits the derivation of a number of ungrammatical sentences, like the following.

(4.11) $*$ Loren Neuromancer read.
(4.12) $*$ Destroyed credibility his Stewart.

These derivations are made possible, simply by the fact that NLP allows us to change the order of *any* two sister formulae in the tree.

LP, which has associativity as well as commutativity, allows us to reorder and rebracket our antecedent formulae in any way we want. While this would allow us to treat languages which have (nearly) free word order, like Latin, even in these languages word order is not completely free. For example, Latin sentences tend to have the preposition occurring closely *before* its argument noun phrase.

What we would like is to have some sort of *controlled* access to the structural rules of associativity and commutativity. We will see a number of solutions to this problem in the next chapter.

4.4 Combinator Calculi for NL

In this section, we will look at three combinator calculi for NL and show that all three are equivalent to the sequent calculus for NL. This has two goals: first, in Section 4.5 we will use one of these calculi for our soundness and completeness result with respect to the Kripke models for NL, and second, we will use one of the other calculi to give a polynomial algorithm in Section 4.6.

The first combinator calculus (Došen, 1988, 1989, 1992) consists of a set of axioms (identity, application and its dual "co-application") and a set of rules (monotonicity for all connectives and transitivity) as shown in Figure 4.5.

Axioms

$$\frac{}{A \vdash A}\ (Id)$$

$$\frac{}{A \bullet (A \backslash B) \vdash B}\ (Appl\backslash) \qquad \frac{}{(B / A) \bullet A \vdash B}\ (Appl/)$$

$$\frac{}{A \vdash B \backslash (B \bullet A)}\ (Co\text{-}appl\backslash) \qquad \frac{}{A \vdash (A \bullet B) / B}\ (Co\text{-}appl/)$$

Rules

$$\frac{A \vdash B \quad C \vdash D}{A \bullet C \vdash B \bullet D}\ (Mon\bullet) \qquad \frac{A \vdash B \quad C \vdash D}{B \backslash C \vdash A \backslash D}\ (Mon\backslash) \qquad \frac{A \vdash B \quad C \vdash D}{C / B \vdash D / A}\ (Mon/)$$

$$\frac{A \vdash B \quad B \vdash C}{A \vdash C}\ (Trans)$$

Fig. 4.5. Došen's axiomatic or combinator presentation of NL

The axiomatic calculus is a bit more tricky to use than either the sequent calculus or natural deduction (try Exercise 4.11 at the end of the chapter to get an idea). This is because the transitivity rule (Trans) plays a role similar to the cut rule in the sequent calculus, but, unlike for the sequent calculus, where the cut rule can be eliminated, the transitivity rule is a necessary component of the axiomatic calculus and finding the intermediate B formula for this rule is not always easy.

4.4.1 Alternative Axiomatic Presentations

Though not directly relevant to the soundness and completeness proofs which follow, it is worthwhile to spend some time discussing two alternative axiomatic presentations of NL, one proposed by Lambek (1988), which is useful to reinforce the links with what we have seen in Chapter 2 — notably the principle of *residuation* — and one proposed by Moortgat and Oehrle (1999), which is a system without the transitivity rules and which we will use in Section 4.6 to prove that we can find NL derivations in polynomial time. Note that when we speak about the axiomatic calculus without further qualification we will mean Došen's formulation of Figure 4.5.

Figure 4.6 shows Lambek's formulation. The only rules in this calculus, besides the (*Axiom*) and (*Trans*) rules are the residuation rules. We will sometimes call this calculus the residuation-based calculus.

Axiom

$$\frac{}{A \vdash A}\,(Id)$$

Rules

$$\frac{B \vdash A \backslash C}{A \bullet B \vdash C}\,(Res_{\backslash\bullet}) \quad \frac{A \bullet B \vdash C}{B \vdash A \backslash C}\,(Res_{\bullet\backslash}) \quad \frac{A \vdash C / B}{A \bullet B \vdash C}\,(Res_{/\bullet}) \quad \frac{A \bullet B \vdash C}{A \vdash C / B}\,(Res_{\bullet/})$$

$$\frac{A \vdash B \quad B \vdash C}{A \vdash C}\,(Trans)$$

Fig. 4.6. Lambek's residuation-based combinatorial presentation

Figure 4.7 shows Moortgat and Oehrle's calculus: it has the application and co-application axioms of the Došen presentation and the residuation rules of the Lambek presentation. This formulation has the important advantage that the (*Trans*) rule is admissible, making the calculus more appropriate for proof search, as we will show in Section 4.6.

Axiom

$$\frac{}{A \vdash A}\,(Id)$$

Rules

$$\frac{A \vdash B \quad C \vdash D}{A \bullet C \vdash B \bullet D}\,(Mon\bullet) \quad \frac{A \vdash B \quad C \vdash D}{B \backslash C \vdash A \backslash D}\,(Mon\backslash) \quad \frac{A \vdash B \quad C \vdash D}{C / B \vdash D / A}\,(Mon/)$$

$$\frac{B \vdash A \backslash C}{A \bullet B \vdash C}\,(Res_{\backslash\bullet}) \quad \frac{A \bullet B \vdash C}{B \vdash A \backslash C}\,(Res_{\bullet\backslash}) \quad \frac{A \vdash C / B}{A \bullet B \vdash C}\,(Res_{/\bullet}) \quad \frac{A \bullet B \vdash C}{A \vdash C / B}\,(Res_{\bullet/})$$

Fig. 4.7. Moortgat & Oehrle's presentation using residuation and monotonicity

Lemma 4.9. *Došen's combinator calculus (Figure 4.5) and Lambek's combination calculus (Figure 4.6) are equivalent.*

Proof. This is fairly easy to see. We show that the different *Res* rules are derived rules of the first calculus and that the *Appl* and *Co-appl* axioms and the monotonicity rules are derivable in the second calculus.

$\Longrightarrow$

We show the *Res* rules are derivable in Došen's axiomatic calculus. We show only the cases for $\backslash$, those for $/$ are symmetric.

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\,(\mathit{Id}) \qquad \dfrac{\vdots}{B \vdash A \backslash C}}{A \bullet B \vdash A \bullet (A \backslash C)}\,(\mathit{Mon}\bullet) \qquad \dfrac{}{A \bullet (A \backslash C) \vdash C}\,(\mathit{Appl}\backslash)}{A \bullet B \vdash C}\,(\mathit{Trans})$$

$$\dfrac{\dfrac{}{B \vdash A \backslash (A \bullet B)}\,(\mathit{Coappl}\backslash) \qquad \dfrac{\dfrac{}{A \vdash A}\,(\mathit{Id}) \qquad \dfrac{\vdots}{A \bullet B \vdash C}}{A \backslash (A \bullet B) \vdash A \backslash C}\,(\mathit{Mon}\backslash)}{B \vdash A \backslash C}\,(\mathit{Trans})$$

$\Longleftarrow$

We first show that *Appl*$\backslash$ and *Co-appl*$\backslash$ are derivable. The cases for $/$ are again symmetric.

$$\dfrac{\dfrac{}{A \backslash B \vdash A \backslash B}\,(\mathit{Id})}{A \bullet (A \backslash B) \vdash B}\,(\mathit{Res}_{\backslash\bullet}) \qquad \dfrac{\dfrac{}{B \bullet A \vdash B \bullet A}\,(\mathit{Id})}{A \vdash B \backslash (B \bullet A)}\,(\mathit{Res}_{\bullet\backslash})$$

To conclude, we show that the monotonicity rules are derived rules of the residuation calculus. We only show $(\mathit{Mon}\bullet)$ and $(\mathit{Mon}\backslash)$, the case for $(\mathit{Mon}/)$ is symmetric to the case for $(\mathit{Mon}\backslash)$.

$$\dfrac{\dfrac{\dfrac{\vdots}{A \vdash B} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdots}{C \vdash D} \qquad \dfrac{\dfrac{}{B \bullet D \vdash B \bullet D}\,(\mathit{Id})}{D \vdash B \backslash (B \bullet D)}\,(\mathit{Res}_{\bullet\backslash})}{C \vdash B \backslash (B \bullet D)}\,(\mathit{Trans})}{B \bullet C \vdash B \bullet D}\,(\mathit{Res}_{\backslash\bullet})}{B \vdash (B \bullet D) / C}\,(\mathit{Res}_{\bullet/})}{A \vdash (B \bullet D) / C}\,(\mathit{Trans})}{A \bullet C \vdash B \bullet D}\,(\mathit{Res}_{/\bullet})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ A \vdash B\end{array} \qquad \dfrac{\dfrac{\dfrac{}{B\backslash C \vdash B\backslash C}\,(Id)}{B\bullet(B\backslash C)\vdash C}\,(Res_{\backslash\bullet})}{B \vdash C/(B\backslash C)}\,(Res_{\bullet/})}{A\vdash C/(B\backslash C)}\,(Trans)}{A\bullet(B\backslash C)\vdash C}\,(Res_{/\bullet}) \qquad \begin{array}{c}\vdots\\ C\vdash D\end{array}}{A\bullet(B\backslash C)\vdash D}\,(Trans)}{B\backslash C\vdash A\backslash D}\,(Res_{\bullet\backslash})$$

This completes the equivalence proof of the two combinator calculi. □

For the equivalence of the third calculus, it is a simple corollary of the proof of Lemma 4.9 and of Lemma 4.14 in the next section (though see Moortgat and Oehrle, 1999, for a direct proof).

Corollary 4.10. *All derivable sequents of the monotonicity-residuation calculus of Moortgat and Oehrle shown in Figure 4.7 are derivable sequents of the two other calculi as well.*

Proof. Since by Lemma 4.9 the two other calculi are equivalent, it suffices to show that the residuation calculus generates all theorems of the residuation-monotonicity calculus. By the proof of Lemma 4.9, the monotonicity rules are admissible in the residuation calculus, therefore all theorems of Moortgat and Oehrle's calculus are theorems of Lambek's calculus. □

4.4.2 Equivalence between the Axiomatic Representation and Sequent Calculus

Since the axiomatic formulation of NL doesn't have the commas and parentheses we used to construct antecedent terms in NL, we introduce a simple translation from antecedent terms of NL to formulae, which replaces all commas by products.

Definition 4.11. *Let $\mathscr{A}$ be an antecedent term, we define the function $\|\mathscr{A}\|^{\bullet}$, which translates an antecedent term into a formula, as follows.*

$\|\mathsf{Lp}\|^{\bullet} = \mathsf{Lp}$
$\|(\mathscr{A},\mathscr{A})\|^{\bullet} = \|\mathscr{A}\|^{\bullet} \bullet \|\mathscr{A}\|^{\bullet}$

Before starting the equivalence proof, we first prove a useful substitution lemma, which shows how we can replace a term B by a less general term A in a context Γ. This lemma is an easy combination of rules $(Mon\bullet)$ and $(Trans)$. The intuition behind the Lemma is that it allows us to strengthen the transitivity rule into something which corresponds to (the translation of) the cut rule.

Lemma 4.12. *If* $\|\Gamma[B]\|^\bullet \vdash C$ *and* $A \vdash B$ *then* $\|\Gamma[A]\|^\bullet \vdash C$

Proof. Assume $A \vdash B$ and $\|\Gamma[B]\|^\bullet \vdash C$. In order to prove $\|\Gamma[A]\|^\bullet \vdash C$, it suffices to show that $A \vdash B$ implies $\|\Gamma[A]\|^\bullet \vdash \|\Gamma[B]\|^\bullet$, since this allows us to combine the hypotheses as follows.

$$\frac{\|\Gamma[A]\|^\bullet \vdash \|\Gamma[B]\|^\bullet \qquad \|\Gamma[B]\|^\bullet \vdash C}{\|\Gamma[A]\|^\bullet \vdash C}\ (Trans)$$

Induction of the length l of the unique path in $\Gamma[A]$ to A (which is of course the same as the path in $\Gamma[B]$ to B).

If $l = 0$ then the context is empty and we have $\Gamma[A] = A$, $\Gamma[B] = B$ and therefore we can conclude that $A \vdash B$ implies $\|\Gamma[A]\|^\bullet \vdash \|\Gamma[B]\|^\bullet$ because they are identical.

If $l > 0$ then, since Γ is not the empty context, Γ is either of the form $(\Delta[B], \Delta')$ or of the form $(\Delta', \Delta[B])$.

If the first case $\|(\Delta[B], \Delta')\|^\bullet = \|\Delta[B]\|^\bullet \bullet \|\Delta'\|^\bullet$. Given that we know by induction hypothesis that $\|\Delta[A]\|^\bullet \vdash \|\Delta[B]\|^\bullet$, we can simply apply the monotonicity rule for the product formula as follows.

$$\frac{\begin{array}{c}\vdots\ IH\\ \|\Delta[A]\|^\bullet \vdash \|\Delta[B]\|^\bullet\end{array} \qquad \dfrac{}{\|\Delta'\|^\bullet \vdash \|\Delta'\|^\bullet}\ (Id)}{\|(\Delta[A], \Delta')\|^\bullet \vdash \|(\Delta[B], \Delta')\|^\bullet}\ (Mon\bullet)$$

The case where Γ is of the form $(\Delta', \Delta[B])$ is symmetric. □

Lemma 4.13. $\Gamma \vdash A$ *is derivable in the sequent calculus iff* $\|\Gamma\|^\bullet \vdash A$ *is derivable in the axiomatic representation.*

Proof

$\Longrightarrow$

We proceed by induction on the length l of the sequent calculus proof.

If $l = 1$ the sequent calculus proof contains a single axiom rule and the axiomatic proof is the same, justified by axiom (Id).

If $l > 1$ then induction hypothesis gives us an axiomatic proof of length $l - 1$ and depending on the last rule, we extend it as follows.

- $(\bullet_h)$ Note that $\|\Gamma[(A,B)]\|^\bullet$ is equal to $\|\Gamma[A \bullet B]\|^\bullet$, so the axiomatic proof we have by induction hypothesis for $\|\Gamma[(A,B)]\|^\bullet$ is a proof of $\|\Gamma[A \bullet B]\|^\bullet$ as well.
- $(\bullet_i)$ Induction hypothesis gives us an axiomatic proof of $\|\Delta\|^\bullet \vdash A$ and an axiomatic proof of $\|\Gamma\|^\bullet \vdash B$, which we can combine into a proof of $\|(\Delta,\Gamma)\|^\bullet \vdash A \bullet B$ as shown below ("*Def* $\|.\|^\bullet$" is not a rule of the calculus, but simply denotes the two antecedent terms translate to the same formula according to Definition 4.11).

$$\cfrac{\cfrac{\begin{array}{c}\vdots\ IH\\ \|\Delta\|^\bullet \vdash A\end{array} \qquad \begin{array}{c}\vdots\ IH\\ \|\Gamma\|^\bullet \vdash B\end{array}}{\|\Delta\|^\bullet \bullet \|\Gamma\|^\bullet \vdash A\bullet B}\,(Mon\bullet)}{\|(\Delta,\Gamma)\|^\bullet \vdash A\bullet B}\,(Def\,\|.\|^\bullet)$$

($\backslash_h$) Induction hypothesis gives us an axiomatic proof of $\|\Gamma[B]\|^\bullet \vdash C$ and of $\|\Delta\|^\bullet \vdash A$. We need to show that $\|\Gamma[(\Delta, A\backslash B)]\|^\bullet \vdash C$.

With $\|\Delta\|^\bullet \vdash A$ (induction hypothesis) and $A\backslash B \vdash A\backslash B$ (*Id*) we apply rule (*Mon*$\bullet$) to obtain $\|\Delta\|^\bullet \bullet A\backslash B \vdash A\bullet(A\backslash B)$. Using axiom (*Appl*$\backslash$) and the transitivity rule (*Trans*) we obtain $\|\Delta\|^\bullet \bullet A\backslash B \vdash B$. Combining this with the other induction hypothesis $\|\Gamma[B]\|^\bullet \vdash C$ using Lemma 4.12 gives us $\|\Gamma[(\Delta\bullet(A\backslash B))]\|^\bullet \vdash C$ which is equivalent to $\|\Gamma[(\Delta, A\backslash B)]\|^\bullet \vdash C$.

The proof schema below displays the different steps used.

$$\cfrac{\cfrac{\cfrac{\cfrac{\begin{array}{c}\vdots\ IH\\ \|\Delta\|^\bullet \vdash A\end{array} \qquad \cfrac{}{A\backslash B\vdash A\backslash B}\,(Id)}{\|\Delta\|^\bullet \bullet A\backslash B \vdash A\bullet(A\backslash B)}\,(Mon\bullet) \qquad \cfrac{}{A\bullet(A\backslash B)\vdash B}\,(Appl\backslash)}{\|\Delta\|^\bullet \bullet A\backslash B\vdash B}\,(Trans)}{\|\Delta\bullet A\backslash B\|^\bullet \vdash B}\,(Def\|.\|^\bullet) \qquad \begin{array}{c}\vdots\ IH\\ \|\Gamma[B]\|^\bullet\vdash C\end{array}}{\cfrac{\|\Gamma[\Delta\bullet A\backslash B]\|^\bullet\vdash C}{\|\Gamma[(\Delta, A\backslash B)]\|^\bullet\vdash C}\,(Def\|.\|^\bullet)}\,Lem\ 4.12$$

($\backslash_i$) Induction hypothesis gives us an axiomatic proof of $\|(A,\Gamma)\|^\bullet \vdash C$, which is equal to $A\bullet\|\Gamma\|^\bullet \vdash C$. Using the axiom (*Id*) to obtain $A\vdash A$ we apply rule (*Mon*$\backslash$) to obtain $A\backslash(A\bullet\|\Gamma\|^\bullet)\vdash A\backslash C$. Now we use axiom (*Co-appl*$\backslash$) to obtain $\|\Gamma\|^\bullet \vdash A\backslash(A\bullet\|\Gamma\|^\bullet)$. We combine this with the previous statement using rule (*Trans*) to obtain $\|\Gamma\|^\bullet\vdash A\backslash C$.

The proof schema below shows the different steps.

$$\cfrac{\cfrac{}{\|\Gamma\|^\bullet\vdash A\backslash(A\bullet\|\Gamma\|^\bullet)}\,(Co\text{-}appl\backslash) \qquad \cfrac{\cfrac{}{A\vdash A}\,(Id) \qquad \cfrac{\begin{array}{c}\vdots\ IH\\ \|(A,\Gamma)\|^\bullet\vdash C\end{array}}{A\bullet\|\Gamma\|^\bullet\vdash C}\,(Def\|.\|^\bullet)}{A\backslash(A\bullet\|\Gamma\|^\bullet)\vdash A\backslash C}\,(Mon\backslash)}{\|\Gamma\|^\bullet\vdash A\backslash C}\,(Trans)$$

($/_h$), ($/_i$) Symmetric to the cases for ($\backslash_h$) and ($\backslash_i$).

(*cut*) Induction hypothesis give us a proof of $\|\Gamma\|^\bullet\vdash A$ and of $\|\Delta[A]\|^\bullet\vdash B$. We apply Lemma 4.12 directly to obtain $\|\Delta[\Gamma]\|^\bullet\vdash B$.

$\Longleftarrow$

This part of the proof is easy, since all rules and axioms have simple proofs in the sequent calculus.

$(Mon\bullet)$

$$\dfrac{\dfrac{\dfrac{\vdots\, IH}{A \vdash B} \qquad \dfrac{\vdots\, IH}{C \vdash D}}{(A,C) \vdash B \bullet D}\,\bullet_h}{A \bullet C \vdash B \bullet D}\,\bullet_i$$

$(Mon\backslash)$

$$\dfrac{\dfrac{\dfrac{\vdots\, IH}{A \vdash B} \qquad \dfrac{\vdots\, IH}{C \vdash D}}{(A, B \backslash C) \vdash D}\,\backslash_h}{B \backslash C \vdash A \backslash D}\,\backslash_i$$

$(Trans)$

$$\dfrac{\dfrac{\vdots\, IH}{A \vdash B} \qquad \dfrac{\vdots\, IH}{B \vdash C}}{A \vdash C}\,cut$$

This completes the equivalence proof of the sequent and the combinator calculus. □

Lemma 4.14. *The residuation-monotonicity calculus of Figure 4.7 is equivalent to the cut-free sequent calculus of NL.*

Proof. We only need to prove that if δ is a cut free NL sequent derivation of $\Gamma \vdash C$, then there is a residuation-monotonicity (ResMon) derivation δ' of $\|\Gamma\|^{\bullet} \vdash C$ which does not use the rule $(Trans)$. The other direction is a direct consequence Corollary 4.10, which shows derivability in the residuation-monotonicity calculus implies derivability in the Došen-style axiomatic calculus, and Lemma 4.13, which shows derivability in the axiomatic calculus implies derivability in the sequent calculus for NL.

We prove the lemma by induction on the length l of δ. If $l = 1$, the we have an axiom rule in the sequent calculus and the (Id) rule in the axiomatic calculus.

If $l > 1$ we do a case analysis on the last rule of the proof.

The only cases which require a little work are when the last rule is $\backslash_h$ or $/_h$. We only sketch the basic idea for the case of $\backslash_h$ here, Exercise 4.12 and 4.14 ask you to give a more formal proof. By induction hypothesis, we have an axiomatic proof of $\|\Gamma[B]\|^{\bullet} \vdash C$. Now, using a series of $Res_{\bullet/}$ and $Res_{\bullet\backslash}$ rules, we can obtain a proof of $B \vdash D$, where D has C as a subformula inside a number of $\backslash$ and $/$ connectives, then we can apply the monotonicity rule for $\backslash$ followed by the residuation rule to change $\backslash$ into $\bullet$, after which we "put back" the context $\Gamma[]$ around the antecedent, by using $Res_{/\bullet}$ and $Res_{\backslash\bullet}$ to obtain $\|\Gamma[(\Delta, A \backslash B)]\|^{\bullet} \vdash C$. This gives us the following schematic proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ IH}{\|\Gamma[B]\|^{\bullet} \vdash C}}{B \vdash D}\ \vdots\ Res_{\bullet/}, Res_{\bullet\backslash} \qquad \dfrac{\vdots\ IH}{\|\Delta\|^{\bullet} \vdash A}}{A \backslash B \vdash \|\Delta\|^{\bullet} \backslash D}\ Mon\backslash}{\|\Delta\|^{\bullet} \bullet A \backslash B \vdash D}\ Res_{\backslash\bullet}$$

$$\vdots\ Res_{/\bullet}, Res_{\backslash\bullet}$$

$$\|\Gamma[(\Delta, A \backslash B)]\|^{\bullet} \vdash C$$

The case for $/_h$ is symmetric.

The other cases are easily checked. In case the last rule is $\backslash_i$, we can simulate it using $Res_{\bullet\backslash}$ as follows.

$$\dfrac{\dfrac{\dfrac{\vdots\ IH}{\|(A,\Gamma)\|^{\bullet} \vdash B}}{A \bullet \|\Gamma\|^{\bullet} \vdash B}\ Def \|.\|^{\bullet}}{\|\Gamma\|^{\bullet} \vdash A \backslash B}\ Res_{\bullet\backslash}$$

Given that $\|\Gamma(A,B)\|^{\bullet}$ is equal to $\|\Gamma[A \bullet B]\|^{\bullet}$, the rule $\bullet_h$ is trivial.

Finally, the rule $\bullet_i$ corresponds to an application of the monotonicity rule for the product as follows.

$$\dfrac{\dfrac{\dfrac{\vdots\ IH}{\|\Delta\|^{\bullet} \vdash A} \qquad \dfrac{\vdots\ IH}{\|\Gamma\|^{\bullet} \vdash B}}{\|\Delta\|^{\bullet} \bullet \|\Gamma\|^{\bullet} \vdash A \bullet B}\ Mon_{\bullet}}{\|(\Delta,\Gamma)\|^{\bullet} \vdash A \bullet B}\ Def_{\|.\|^{\bullet}}$$

□

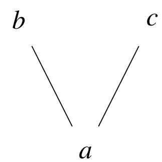
4.5 Model Theory

We have seen group models for the Lambek calculus in the previous chapter. Even though it is possible to adapt these models to the non-associative Lambek calculus, we will introduce a different kind of models for the non-associative Lambek calculus in this chapter: Kripke models. This is another important group of models for the Lambek calculus and they have the advantage of easily accommodating the multimodal extensions to NL we will see in the next chapter.

Kripke models were originally introduced for modal logics and they used possible worlds and a binary accessibility relation R^2 between worlds to give models for logical necessity and possibility (for many other applications and a very good general overview of modal logics, see Blackburn et al, 2001).

For our current purposes, however, our ‘worlds’ are simply the linguistic structures we use: formulae and their (structured) combinations. In the following text we will use the words ‘world’ and ‘linguistic resource’ interchangeably.

We use a *ternary* accessibility relation R^3 to “merge” these formulae and structures: R^3abc holds between a b and c if and only if resource a is the result of merging resource b and resource c. We can represent R^3abc in the form of a picture as shown below.



We intend to interpret this ternary relation in such a way that if, for example, b is in the interpretation of np and c is in the interpretation of $np\backslash S$ then we want to conclude that there exists a world a such that a is in the interpretation of S.

Though it is often useful to draw the accessibility relation as a tree-like structure, as shown above, it is important to remember that the structures we define using this ternary relation do not necessarily represent trees. In particular, if we want to interpret R^3abc as saying that a is the mother of its daughters b and c, we have to keep in mind that unicity of this mother for a given b and c is *not* necessary, or, more precisely, the formula $\forall a \forall b \forall c \forall d.(R^3abc \wedge R^3dbc) \rightarrow d = a$ does not hold. However, to make the intuitions behind the use of the accessibility relation R^3 clear, we will sometime refer to a trio of nodes a, b, c such that R^3abc using the vocabulary of binary trees, saying the a is the parent of b and c or that b is the left sister of c.

Before giving the definitions of the connectives, we first give some standard definitions in modal logic: frames, models, etc. In what follows, we will assume the set of atomic formulae P to be fixed.

Definition 4.15. *A* Kripke frame *F is a pair $\langle W, R^3 \rangle$ where W is a non-empty set of worlds and R^3 is a ternary accessibility relation over triples of elements from W.*

Definition 4.16. *A* Kripke model *M is a triple $\langle W, R^3, V \rangle$ where $\langle W, R^3 \rangle$ is a Kripke frame and where V is an evaluation function from elements of P to subsets of W. We say that $\langle W, R^3 \rangle$ is the* underlying frame *of M.*

The following definitions are also standard.

Definition 4.17. *A sequent $A \vdash B$ is* true in a model M at world a *— we will write $M, a \models (A \vdash B)$ — in case if $M, a \models A$ then $M, a \models B$ as well.*

A sequent $A \vdash B$ *is* true in a model M *— we will write* $M \models (A \vdash B)$ *— iff it is true at all worlds in that model.*

A sequent $A \vdash B$ *is* valid on a frame F *— we will write* $F \models (A \vdash B)$ *— iff it is true at all worlds and under all valuations of that frame.*

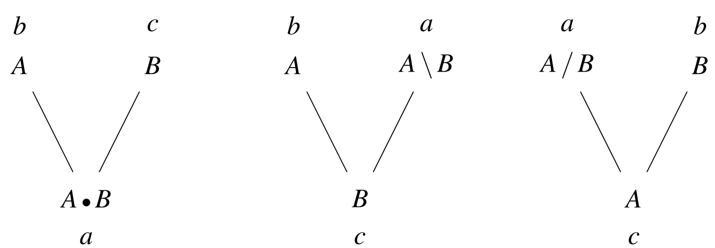
The notion of Kripke frame is often useful when we want to study properties of the accessibility relation without considering the assignments of the atomic formulae. We will return to this point in Section 4.5.2, where we see that adding structural rules to NL corresponds to restricting the frame we use for interpreting the logic.

As is clear from Definition 4.16, a Kripke model is simply a Kripke frame with an evaluation which maps atomic formulae to sets of worlds in our model, so, for example, it would tell us at which worlds the atomic formula np is true and at which worlds the atomic formula S is true. The interpretation of the atomic formulae can vary from one model to another.

So, though the interpretation of the atomic formulae is fixed by the model, the interpretation of the complex formulae defined as follows.

$$\begin{array}{rl} M,a \models p & \textit{iff}\ a \in V(p) \\ M,a \models A \bullet B & \textit{iff}\ \exists b \exists c (R^3 abc \,\&\, M,b \models A \,\&\, M,c \models B) \\ M,a \models A \backslash B & \textit{iff}\ \forall b \forall c ((R^3 cba \,\&\, M,b \models A) \Rightarrow M,c \models B) \\ M,a \models A / B & \textit{iff}\ \forall b \forall c ((R^3 cab \,\&\, M,b \models A) \Rightarrow M,c \models B) \end{array}$$

Again, the picture below helps us to better see the different worlds, the formulae which hold at them and the way they are related by the accessibility relation.



Seen this way, the semantics of the connectives is quite close to their intuitive meanings: if $A \bullet B$ holds at world a then this world is the merger by R^3abc of two worlds b and c such that A holds at world b and B holds at world c. Similarly, if A / B holds at world a then for any world b which is "to the right" of a via R^3cab and which is in the interpretation of B, then the world c, which represents the "parent" or the merger of a and b, is in the interpretation of A.

4.5.1 Soundness and Completeness

Lemma 4.18 (Soundness). *If* $A \vdash B$ *is derivable then for all models* M *and for all words* a, $M,a \models (A \vdash B)$.

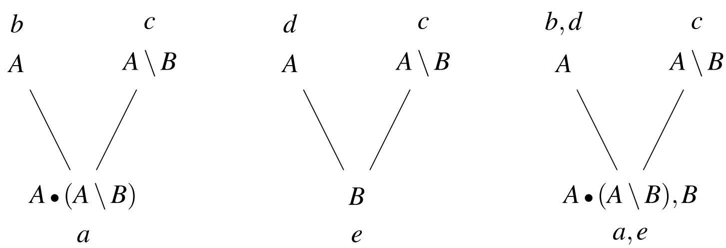
Proof. Soundness is relatively simple. We prove by induction on the depth of the axiomatic proof of $A \vdash B$ that for all models M and worlds a, whenever $M,a \models A$ then $M,a \models B$. This is just a matter of writing out the definitions for the different connectives.

First, we look at the axioms.

(Id) Axiom (Id) is a simple tautology.

($Appl\backslash$) Suppose we obtained $A \bullet (A \backslash B) \vdash B$ by application of axiom ($Appl\backslash$). In that case, we need to prove that for every world a such that $M,a \models A \bullet (A \backslash B)$, B is true at world a in M as well. Given the definition of the product, this means there are worlds b and c such that R^3abc and $M,b \models A$ and $M,c \models A \backslash B$. The picture below on the left shows these three worlds and their relation.
We can then unfold the definition of $A \backslash B$ at world c, obtaining that for every world d and e such that R^3edc and $M,d \models A$, $M,e \models B$ as well. But if this holds for any worlds, then it also holds for $d = b$, for which we already knew that $M,b \models A$ and for world $e = a$, which means R^3edc holds by virtue of being equal to R^3abc. $M,b \models A$ and R^3abc together then imply $M,a \models B$ which we needed to prove.

The picture below shows the unfolding of the product formula on the left, then the unfolding of the implication in the middle and finally the composed figure on the right.



($Appl/$) Symmetric.

($Co\text{-}appl\backslash$) If we obtained $A \vdash B \backslash (B \bullet A)$ as an instantiation of axiom ($Co\text{-}appl\backslash$), we verify that for every model M and world a, if $M,a \models A$ then $M,a \models B \backslash (B \bullet A)$, that is, we verify that for all worlds b and c such that R^3cba and $M,b \models B$ we have $M,c \models B \bullet A$. Take any worlds b and c such that R^3cba and $M,b \models B$. We need to show that $M,c \models B \bullet A$. In other words, we need to show that there exists a d and an e such that R^3cde and $M,d \models B$ and $M,e \models A$. Take $b = d$ and $a = e$, then $R^3cde = R^3cba$, $M,b \models B$ and $M,a \models A$ all hold by assumption. Therefore, we have shown that if $M,a \models A$ then $M,a \models B \backslash (B \bullet A)$.

($Co\text{-}appl/$) Symmetric.

Next for the rules. For rules ($Mon\bullet$), ($Mon\backslash$) and ($Mon/$) we know by induction hypothesis that for all models M and worlds a, if $M,a \models A$ then $M,a \models B$ and if $M,a \models C$ then $M,a \models D$.

($Mon\bullet$) For rule ($Mon\bullet$) we have to show that for every model M and world a if $M,a \models A \bullet C$ then $M,a \models B \bullet D$. Take an arbitrary world and model satisfying $M,a \models A \bullet C$. The definition of $A \bullet C$ tells us there are worlds b and c such that R^3abc, $M,b \models A$ and $M,c \models C$. Induction hypothesis gives us that $M,b \models B$ and $M,c \models D$ and then, given that we already knew R^3abc we can apply the definition of the product in the opposite direction to obtain $M,a \models B \bullet D$.

($Mon\backslash$) We have to show that for every model M and world a, if $M,a \models B \backslash C$ then $M,a \models A \backslash D$. Given that $M,a \models B \backslash C$ we know that for all worlds b and c if R^3cba and $M,b \models B$ then $M,c \models C$. But induction hypothesis also gives us $M,b \models A$ and $M,c \models D$ for all b and c, allowing us to apply the definition of $\backslash$ again to obtain $M,a \models A \backslash D$.

($Mon/$) Symmetric.

(*Trans*) Induction hypothesis gives us that for all models M and worlds a, if $M,a \models A$ then $M,a \models B$ and if $M,a \models B$ then $M,a \models C$. We just have to show that for every model M and every world a if $M,a \models A$ then $M,a \models C$. Suppose $M,a \models A$. By induction hypothesis we know that $M,a \models A$ implies $M,a \models B$. Induction hypothesis also gives us that $M,a \models B$ implies $M,a \models C$, which is all we needed to show. □

For the completeness result, we first define a canonical model for NL and prove a strong result: the Truth Lemma which states that truth in this model corresponds exactly with provability in the axiomatic calculus. This means that for *any* underivable statement $A \vdash B$, our canonical model will be a countermodel but also that for any statement $A \vdash B$ which is true in the canonical model, $A \vdash B$ is derivable.

Definition 4.19. *The canonical model* $\mathscr{M} = \langle W, R^3, V \rangle$ *for NL is defined as follows:*

- *W is the set of all formulae,*
- *R^3abc iff $a \vdash b \bullet c$, and*
- *$a \in V(p)$ iff $a \vdash p$.*

Let's look at the different clauses for the canonical model in a bit more detail. First, the worlds of the canonical model are just the formulae of NL. Second, the accessibility relation is linked directly to the derivability relation of the product formula. Finally, for all atomic formulae p and all formulae (worlds) a such that $a \vdash p$ we have $a \in V(p)$. In other words, $V(p)$ is true at world p and at all words a such that we can derive p from a.

Lemma 4.20. *The* Truth Lemma*, $\mathscr{M}, a \models A$ iff $a \vdash A$*

Proof.

$\Longleftarrow$

Soundness is a simple corollary of Lemma 4.18 which proves the stronger claim that for *any* model M and word a we have $M,a \models A$ implies $a \vdash A$.

$\Longrightarrow$

For the completeness part we show that if $\mathscr{M}, a \models A$ then $a \vdash A$. In other words, when $a \models A$ is true in the canonical model for formulae a and A, then there is a derivation of $a \vdash A$ in the axiomatic calculus of Figure 4.5 as well. We prove the completeness part of the Truth Lemma by induction on A.

[$A = p$] For atomic formulae p the definition of $\models$ for $\mathscr{M}$ gives us $a \vdash p$ directly, by construction.

[$A = B \bullet C$] A is of the form $B \bullet C$. Assume that a is a world such that $\mathscr{M}, a \models B \bullet C$. By the definition of $M, a \models B \bullet C$ there are worlds b and c such that R^3abc where $\mathscr{M}, b \models B$ and $\mathscr{M}, c \models C$. Given that B and C are subformulae of A, we can apply the induction hypothesis and obtain proofs $b \vdash B$ and $c \vdash C$. Using rule (*Mon*$\bullet$) we obtain

$$b \bullet c \vdash B \bullet C$$

But, given that R^3abc, the definition of R^3 in the canonical frame gives us.

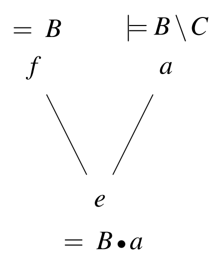
$$a \vdash b \bullet c$$

Finally, using the transitivity rule (*Trans*) we obtain

$$a \vdash B \bullet C$$

as required.

[$A = B \backslash C$] Assume that a is a world such that $\mathscr{M}, a \models B \backslash C$, we have to show $a \vdash B \backslash C$. Given that $\mathscr{M}, a \models B \backslash C$ this means that for all e and f if R^3efa and $\mathscr{M}, f \models B$ then $\mathscr{M}, e \models C$. Choose $e = B \bullet a$ and $f = B$. The picture below summarizes the worlds we have named so far and their relations.



Now we can use the fact that the derivability of $B \bullet a \vdash B \bullet a$ (which is equal to $e \vdash f \bullet a$) and the definition of R^3 in the canonical frame give us R^3efa. Given that $\mathscr{M}, f \models B$ and R^3efa are satisfied, the definition of $B \backslash C$ gives us $\mathscr{M}, e \models C$, with $e = B \bullet a$. By induction hypothesis, we have a proof of $B \bullet a \vdash C$. We can transform this proof into a proof of $a \vdash B \backslash C$ as follows.

$$\dfrac{\dfrac{}{a \vdash B\backslash(B\bullet a)}\,(Co\text{-}appl\backslash) \qquad \dfrac{\dfrac{}{B\vdash B}\,(Id) \qquad \dfrac{\vdots\, IH}{B\bullet a \vdash C}}{B\backslash(B\bullet a)\vdash B\backslash C}\,(Mon\backslash)}{a \vdash B\backslash C}\,(Trans)$$

$[A = C/B]$ Symmetric. □

Corollary 4.21. *An important corollary of the Truth Lemma is that the canonical model $\mathscr{M}$ is the* most general model *for NL, that is, for every model M if $\mathscr{M} \models A \vdash B$ then $M \models A \vdash B$.*

Proof. The corollary has a very easy proof by contraposition. We show that if there is a model M such that $M \not\models A \vdash B$ then $\mathscr{M} \not\models A \vdash B$. But if the statement $A \vdash B$ has a countermodel then $A \nvdash B$ (the sequent is underivable), which by the Truth Lemma means $\mathscr{M} \not\models A \vdash B$. □

Theorem 4.22. *NL is sound and complete with respect to all models.*

Proof. Soundness was proved as Lemma 4.18 and with the Truth Lemma in place, the completeness proof is trivial.

For completeness, we need to show that if a statement $A \vdash B$ is true in all models, then $A \vdash B$ is derivable. But if $A \vdash B$ is true in all models, then it is true in the canonical model as well, so we have $\mathscr{M} \models (A \vdash B)$. In other words, for any world a we have that if $\mathscr{M}, a \models A$ then $\mathscr{M}, a \models B$ (1). If this holds for any world a then in particular for world A, and since $A \vdash A$ is derivable, by the Truth Lemma we have $\mathscr{M}, A \models A$ and therefore $\mathscr{M}, A \models B$, by (1). Now by the Truth Lemma $\mathscr{M}, A \models B$ implies that $A \vdash B$ is derivable. □

4.5.2 Adding Structural Rules

An interesting benefit of the current formulation is that, like in modal logic, we can add *restrictions* to our frame to change the properties of the connectives in our logic. This branch of modal logic is called *correspondence theory*: a well-know example is that the modal logic S4 corresponds to the reflexive, transitive frames. Chapter 3 of Blackburn et al (2001) gives a detailed overview of correspondence theory for modal logic.

Many of the standard techniques which apply to frames for modal logics can be adapted to categorial grammars. In this section we will be interested in finding first-order formulae which express constraints on modal frames (see Došen, 1992; Kurtonina, 1995, 1998). Kurtonina (1995, 1998) shows that first-order constraints allow us to define a large class of categorial logics; we will return to her results briefly in Section 5.5.1.

For now, we will only treat the structural rules of associativity and commutativity, as we have seen them in Section 4.3.

The structural rules for associativity and commutativity can be added to our models by adding the following restrictions on the accessibility relation R^3.

$$\forall a \forall b \forall c.(R^3abc \Rightarrow R^3acb) \quad \text{(com)}$$
$$\forall a \forall b \forall c \forall d \forall f.((R^3afd \mathbin{\&} R^3fbc) \Rightarrow \exists e.(R^3abe \mathbin{\&} R^3ecd)) \quad \text{(ass1)}$$
$$\forall a \forall b \forall c \forall d \forall e.((R^3abe \mathbin{\&} R^3ecd) \Rightarrow \exists f.(R^3afd \mathbin{\&} R^3fbc)) \quad \text{(ass2)}$$

If we want to add several of these constraints to the accessibility relation, this corresponds simply to adding the conjunction of the corresponding formula as a frame constraint.

As usual, these principles are best shown in picture form, as done in Figure 4.8.

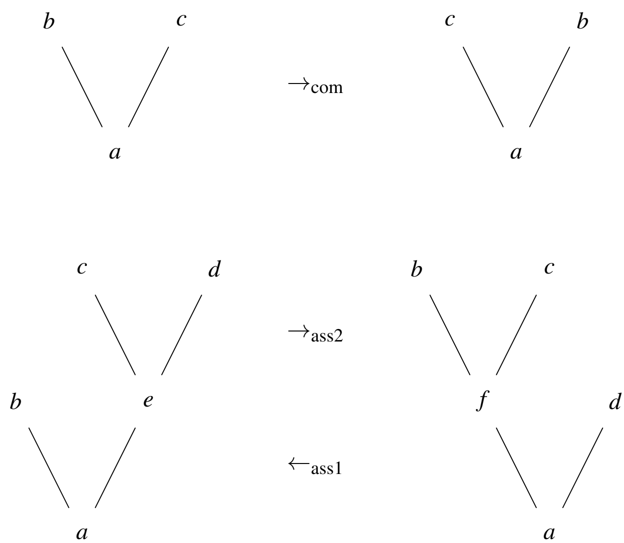


Fig. 4.8. Visual representation of the constraints on the accessibility relation R^3 corresponding to associativity and commutativity

These frame constraints correspond to adding the following axioms to the axiomatic calculus for NL.

$$A \bullet B \vdash B \bullet A \quad \text{(com)}$$
$$A \bullet (B \bullet C) \vdash (A \bullet B) \bullet C \quad \text{(ass1)}$$
$$(A \bullet B) \bullet C \vdash A \bullet (B \bullet C) \quad \text{(ass2)}$$

Now, the following is easy to see.

Proposition 4.23. *The axiomatic calculus with a subset of the additional axioms* $\{$ *(comm), (ass1), (ass2)*$\}$ *corresponds to the sequent calculus with the additional structural rules of the same name (these are shown in Figure 4.3)*

Proof. This is an easy extension of Lemma 4.13. We show only the case for (ass1), the other cases are similar.

Showing that $A \bullet (B \bullet C) \vdash (A \bullet B) \bullet C$ is derivable in the sequent calculus using the structural rule (ass1) of Figure 4.3 (repeated below) is trivial.

$$\frac{\Gamma[((\Delta_1,\Delta_2),\Delta_3)] \vdash D}{\Gamma[(\Delta_1,(\Delta_2,\Delta_3))] \vdash D} \text{(ass1)}$$

Now, for the inverse direction, we only need to treat the new case (the rest of the proof follows from Lemma 4.13). Suppose we have an axiomatic proof of $\|\Gamma[((\Delta_1,\Delta_2),\Delta_3)]\|^{\bullet} \vdash D$, then we need to show we can transform it into an axiomatic proof of $\|\Gamma[(\Delta_1,(\Delta_2,\Delta_3))]\|^{\bullet} \vdash D$. By the definition of $\|.\|^{\bullet}$, this is equivalent to showing we can transform a proof of $\|\Gamma[(\Delta_1 \bullet \Delta_2) \bullet \Delta_3]\|^{\bullet} \vdash D$ into a proof of $\|\Gamma[\Delta_1 \bullet (\Delta_2 \bullet \Delta_3)]\|^{\bullet} \vdash D$.

According to Lemma 4.12, whenever $\|\Gamma[F]\|^{\bullet} \vdash D$ and $E \vdash F$ are derivable in the axiomatic calculus, then $\|\Gamma[E]\|^{\bullet} \vdash D$ is derivable as well. Taking $E = \|\Delta_1\|^{\bullet} \bullet (\|\Delta_2\|^{\bullet} \bullet \|\Delta_3\|^{\bullet})$ and $F = (\|\Delta_1\|^{\bullet} \bullet \|\Delta_2\|^{\bullet}) \bullet \|\Delta_3\|^{\bullet}$, makes $E \vdash F$ an instantiation of the axiom (ass1) and gives us $\|\Gamma[(\Delta_1 \bullet \Delta_2) \bullet \Delta_3]\|^{\bullet} \vdash D$ implies $\|\Gamma[\Delta_1 \bullet (\Delta_2 \bullet \Delta_3)]\|^{\bullet} \vdash D$ as required. □

Lemma 4.24. *Let F be a frame and FO be the first-order formula expressing one of the frame constraints { (com), (ass1), (ass2) } and $A \vdash B$ the corresponding sequent. F satisfies FO if and only if $F \models (A \vdash B)$.*

Proof
We verify only the case for (com), the cases for associativity are similar.

For (com) this means we need to show that F satisfies $\forall a \forall b \forall c.(R^3abc \Rightarrow R^3acb)$ iff $F \models (A \bullet B \vdash B \bullet A)$.

$\Longrightarrow$

We need to show that if F satisfies the frame constraint, then F satisfies $A \bullet B \vdash B \bullet A$ as well. Let M be any model $\langle W, R^3, V\rangle$ such that its underlying frame $F = \langle W, R^3\rangle$ satisfies the constraint $\forall a \forall b \forall c.(R^3abc \Rightarrow R^3acb)$ and let d be any world in this model. Suppose $M,d \models A \bullet B$. We need to show that $M,d \models B \bullet A$. If $M,d \models A \bullet B$ then writing out the definition of the valuation for $A \bullet B$ we conclude that there are worlds e and f in the model such that R^3def, $M,e \models A$ and $M,f \models B$. Applying the frame constraint with $a = d$, $b = e$ and $c = f$ gives us R^3dfe. Therefore, at world d the following are true: R^3dfe, $M,e \models A$ and $M,f \models B$. This means that $M,d \models B \bullet A$ as required.

$\Longleftarrow$

We prove the other direction by contraposition. Suppose our frame does *not* satisfy the frame constraint, that is F satisfies its negation, $\exists c \exists a \exists b.(R^3cab \,\&\, \neg R^3cba)$. In other words, there are worlds a, b and c such that R^3cab but not R^3cba. Given this frame F, we can construct a countermodel by giving a valuation v making $A \bullet B \vdash B \bullet A$ invalid: set $v(A) = \{a\}$, $v(B) = \{b\}$. We claim that $M,c \models A \bullet B$ but that $M,c \not\models B \bullet A$. In order to show that $A \bullet B$ is true at c, we need to show that

$\exists x \exists y.R^3cxy \,\&\, M,x \models A \,\&\, M,y \models B$. Choosing $x = a$ and $y = b$ makes this statement true by our chosen validation. What remains to be done is show that $M,c \not\models B \bullet A$, that is $\neg\exists x \exists y.(R^3cxy \,\&\, M,x \models B \,\&\, M,y \models A)$. We prove this by contradiction: suppose there are x and y which satisfy the three conjuncts. Since $M,x \models B$ and $M,y \models A$ our valuation forces $x = b$ and $y = a$. This means R^3cba must hold as well, contradicting $\neg R^3cba$ from our assumption that F does not satisfy the frame constraint. □

The important point of this section is that Kripke models give fairly easy soundness and completeness results not just for the non-associative Lambek calculus but also — with the appropriate frame constraints, for the Lambek calculus L — as well as for the commutative versions both calculi, NLP and LP.

4.6 Polynomial Complexity

In this section, we will talk about the complexity of parsing sentences and proving theorems for the non-associative Lambek calculus. It is important in this context to distinguish between the complexity of theorem proving and the complexity of parsing: theorem proving asks the question whether a given sequent is derivable in a calculus, whereas parsing asks the question of whether, given a sentence $w_1,\dots,w_n$ and a lexicon Lex mapping words to formulas, there is a sequence $f_1,\dots f_n$ such that each $f_i \in \mathrm{Lex}(w_i)$ and an antecedent term Γ with yield $f_1,\dots,f_n$ such that $\Gamma \vdash S$ is derivable. From this, it is easy to see that parsing is at least as difficult as theorem proving.

4.6.1 Complexity

Before talking about NL, we will very briefly discuss some of the known results for other Lambek calculi: the associative Lambek calculus L and the associative, commutative Lambek-van Benthem calculus LP. We have a rather complete picture of the complexity of the different Lambek calculi NL, L and LP, both with the product $\bullet$ and without it.

Figure 4.9 summarizes the results. For LP, its relationship with linear logic, which we will discuss in more detail in Chapter 6, makes it possible to apply the complexity results for the multiplicative fragment of linear logic (Kanovich, 1994) directly to LP to obtain NP completeness results, both for the logic which contains only implication and for the logic with implication and conjunction. NP-completeness for the Lambek calculus with product was proved by Pentus (2006); Savateev (2009) showed that NP-completeness holds even in the case without the product formula. All these results are for theorem proving, but it is easy to see that the parsing problem has the same complexity: given that parsing is at least as difficult as theorem proving and that theorem proving is NP complete, we only need to show that parsing is in NP to show NP-completeness and it is as easy to verify whether or not a parse is successful as it it to verify whether or not a proof is valid.

Though L and LP are NP complete both for parsing and for theorem proving, there are polynomial algorithms for NL. However, there is an important difference between NL with product, for which we show *theorem proving* is polynomial and product-free NL, for which *parsing* is polynomial. We will discuss these polynomial algorithms in what follows.

Polynomial	*NP complete*	
$\mathrm{NL}^{\backslash/\bullet}$ de Groote (1999)	$\mathrm{L}^{\backslash/\bullet}$ Pentus (2006)	$\mathrm{LP}^{\backslash/\bullet}$ Kanovich (1994)
$\mathrm{NL}^{\backslash/}$ Aarts and Trautwein (1995)	$\mathrm{L}^{\backslash/}$ Savateev (2009)	$\mathrm{LP}^{\backslash/}$ Kanovich (1994)

Fig. 4.9. The complexity of different variants of the Lambek calculus

4.6.2 De Groote's Context Calculus SC

Philippe de Groote has shown that theorem proving for NL takes polynomial time (de Groote, 1999), generalizing an earlier result from Aarts and Trautwein (1995) for product-free NL which we will discuss in Section 4.6.4. In this section, we will first present de Groote's result and then talk about the special case without product.

The context calculus SC is defined using two-formula sequents and *contexts* which are defined as follows.

Definition 4.25. *A formula with a hole or a* context *is defined as follows:*

$$\mathscr{F}[] \quad ::= \quad [] \mid \mathscr{F}[] \backslash \mathsf{Lp} \mid \mathsf{Lp} \backslash \mathscr{F}[] \mid \mathscr{F}[] / \mathsf{Lp} \mid \mathsf{Lp} / \mathscr{F}[] \mid \mathscr{F}[] \bullet \mathsf{Lp} \mid \mathsf{Lp} \bullet \mathscr{F}[]$$

Though similar in spirit to the definition of context in the beginning of this chapter (Definition 4.4), where a context is an antecedent term with a hole as opposed to a formula with a hole here, the two notions are of course distinct. We will use $\Gamma[], \Delta[], \Theta[], \ldots$ to range over contexts. $\Gamma[A]$ is the formula obtained by filling the hole in $\Gamma[]$ by A.

The context calculus SC is shown in Figure 4.10.

As can be seen in the rules, we need to distinguish between positive and negative contexts. The context rules $\bullet / N$ and $\bullet \backslash N$ should be compared — modulo some manipulation of contexts — to the (*Appl*) axioms of the axiomatic presentation of Figure 4.5 on page 113, the $/ \bullet P$ and $\backslash \bullet P$ rules correspond to the (*Co-appl*) axioms, whereas the $/ \backslash P$ and $\backslash / P$ rules correspond to the type lifting rules.

Sequent Rules

$$\frac{}{A \vdash A}\,\mathit{axiom}$$

$$\frac{A \vdash B \quad C \vdash D}{A \bullet C \vdash B \bullet D}\,\bullet mon \qquad \frac{A \vdash B \quad C \vdash D}{B \backslash C \vdash A \backslash D}\,\backslash mon \qquad \frac{A \vdash B \quad C \vdash D}{A / D \vdash B / C}\,/mon$$

$$\frac{A \vdash B \quad \vdash_N \Gamma[\,]}{\Gamma[A] \vdash B}\,cont_N \qquad \frac{A \vdash B \quad \vdash_P \Gamma[\,]}{A \vdash \Gamma[B]}\,cont_P$$

Negative Context Rules

$$\frac{}{\vdash_N [\,]}\,[\,]N$$

$$\frac{A \vdash B \quad \vdash_N \Gamma[\,] \quad \vdash_N \Delta[\,]}{\vdash_N (A \bullet \Gamma[(B \backslash \Delta[\,])])}\,\bullet\backslash N \qquad \frac{A \vdash B \quad \vdash_N \Gamma[\,] \quad \vdash_N \Delta[\,]}{\vdash_N (\Gamma[(\Delta[\,] / B)] \bullet A)}\,\bullet/N$$

Positive Context Rules

$$\frac{}{\vdash_P [\,]}\,[\,]P$$

$$\frac{A \vdash B \quad \vdash_P \Gamma[\,] \quad \vdash_P \Delta[\,]}{\vdash_P (A \backslash \Gamma[(B \bullet \Delta[\,])])}\,\backslash\bullet P \qquad \frac{A \vdash B \quad \vdash_P \Gamma[\,] \quad \vdash_P \Delta[\,]}{\vdash_P (\Gamma[(\Delta[\,] \bullet B)] / A)}\,/\bullet P$$

$$\frac{B \vdash A \quad \vdash_N \Gamma[\,] \quad \vdash_P \Delta[\,]}{\vdash_P (A / \Gamma[(\Delta[\,] \backslash B)])}\,/\backslash P \qquad \frac{B \vdash A \quad \vdash_N \Gamma[\,] \quad \vdash_P \Delta[\,]}{\vdash_P (\Gamma[(B / \Delta[\,])] \backslash A)}\,\backslash/P$$

Fig. 4.10. The context calculus SC from de Groote (1999)

Before analyzing the complexity of proof search in the calculus SC, we first need to show that it is equivalent to NL. That is. we need to show that SC derives all and only the theorems of NL. To do this, we follow the structure of the proof of de Groote (1999) and first introduce some auxiliary notions.

Definition 4.26. *A negative context* $\Gamma[]$ *is* correct *iff* $A \vdash B$ *implies* $\Gamma[A] \vdash B$.
A positive context $\Delta[]$ *is* correct *iff* $A \vdash B$ *implies* $A \vdash \Delta[B]$.

The following lemma shows that we can "nest" contexts.

Lemma 4.27. *If* $\vdash_N \Gamma[]$ *and* $\vdash_N \Delta[]$ *are derivable using the negative context rules, then* $\vdash_N \Gamma[\Delta[]]$ *is derivable as well.*

If $\vdash_P \Gamma[]$ *and* $\vdash_P \Delta[]$ *are derivable using the positive context rules, then* $\vdash_P \Gamma[\Delta[]]$ *is derivable as well.*

Proof. Both parts of the lemma are proved by a simple induction on $\vdash_N \Gamma[]$ resp. $\vdash_P \Gamma[]$. □

For the completeness proof, it is useful to restrict the SC derivation into those that have a certain form.

Definition 4.28. *An SC derivation is* normal *if the following conditions are satisfied.*

1. *The rule axiom is restricted to atomic formulae.*
2. *No rule* $cont_N$ *or* $cont_P$ *has an axiom* $[\,]N$ *resp.* $[\,]P$ *as its right premise.*
3. *No rule* $cont_N$ *or* $cont_P$ *has another rule* $cont_N$ *or* $cont_P$ *as its left premise.*

Lemma 4.29. *Every SC derivation can be transformed into a normal SC derivation.*

Proof. Condition 1 corresponds to the possibility to restrict NL to atomic axioms. In the calculus SC, the proof is even easier than in the sequent calculus, and follows directly from the monotonicity rules for the three connectives of the calculus.

Condition 2 is trivial as well. If the right premise on a *cont* rule is a context axiom, then the context is empty and the conclusion of the rule is equal to the left premise and the *cont* rule can be eliminated from the proof.

Condition 3, finally, is a direct corollary of Lemma 4.27. □

Lemma 4.30. *SC is equivalent to the monotonicity-residuation presentation of NL shown in Figure 4.7 on page 114.*

Proof. We show that a sequent is derivable in the context calculus SC if and only if the corresponding sequent is derivable in the combinatorial presentation ResMon. Our work is made easier by the fact that the two calculi share the monotonicity and axiom rules.

$\Longrightarrow$

We show by induction on the length of the SC proof of $A \vdash B$ that there is a derivation in the residuation-monotonicity calculus of $A \vdash B$ as well.

The *axiom* rule and the three *mon* rules are identical to the *Id* and monotonicity rules of the residuation-monotonicity calculus.

The negative and positive context rules are valid by the definition of correct contexts.

So, what remains to be shown is that the context rules allow us to derive only correct contexts.

Negative contexts

For a negative context $\Gamma[]$, being correct means that if $A \vdash B$, then $\Gamma[A] \vdash B$.

- If $\vdash_N \Gamma[]$ is obtained by the axiom for negative contexts $[\,]N$, the result holds trivially.
- If $\vdash_N \Theta[]$ is obtained by the $\bullet\backslash N$ rule, then Θ is of the form $A \bullet \Gamma[(B \backslash \Delta[])]$ and, by induction hypothesis, we have a ResMon proof of $A \vdash B$ and we know that $\Gamma[]$ and $\Delta[]$ are correct negative contexts. We need to show that $\Theta[]$ is a correct negative context as well, that is, if $C \vdash D$ then $A \bullet \Gamma[(B \backslash \Delta[C])] \vdash D$.

  Now, if $C \vdash D$, then, given that $\Delta[]$ is a correct negative context by induction hypothesis, we can conclude $C \vdash \Delta[D]$. $C \vdash \Delta[D]$ and $A \vdash B$ (induction hypothesis) together allow us to conclude $B \backslash \Delta[C] \vdash A \backslash D$ by $(Mon\backslash)$. This, together with the fact that $\Gamma[]$ is a correct positive context, allows us to conclude $\Gamma[(B \backslash \Delta[C])] \vdash A \backslash D$, followed by $(Res_{\backslash\bullet})$ to conclude $A \bullet \Gamma[(B \backslash \Delta[C])] \vdash D$ as required.

$$\dfrac{\dfrac{\dfrac{\stackrel{\vdots\, IH}{A \vdash B} \qquad \dfrac{C \vdash D}{\Delta[C] \vdash D}\,\Delta[]\ correct}{B \backslash \Delta[C] \vdash A \backslash D}\,(Mon_\backslash)}{\Gamma[(B \backslash \Delta[C])] \vdash A \backslash D}\,\Gamma[]\ correct}{A \bullet \Gamma[(B \backslash \Delta[C])] \vdash D}\,(Res_{\backslash\bullet})$$

- The case for $\bullet / N$ is symmetric.

Positive contexts

If $\Delta[]$ is a positive context such that $\vdash_P \Delta[]$ and $A \vdash B$ is derivable in ResMon, then $A \vdash \Delta[B]$ is derivable in ResMon as well. We use induction on the proof of $\vdash_P \Delta[]$.

- If $\vdash_P \Delta[]$ is obtained by the axiom for negative contexts $[\,]P$, the result holds trivially.
- If $\vdash_P \Theta[]$ is obtained by the $\backslash\bullet P$ rule, then $\Theta[]$ is of the form $A \backslash \Gamma[(B \bullet \Delta[])]$, we know by induction hypothesis that $\Gamma[]$ and $\Delta[]$ are correct positive contexts and that $A \vdash B$ is derivable. In order to show that $\Theta[]$ is a correct context, we need to show that if $C \vdash D$ then $C \vdash A \backslash \Gamma[(B \bullet \Delta[D])]$. Given that we know by induction hypothesis that $\Delta[]$ is a correct positive context, $C \vdash D$ allows us to conclude $C \vdash \Delta[D]$, which together with $A \vdash B$ (induction hypothesis) gives us $A \bullet C \vdash B \bullet \Delta[D]$ using $Mon_\bullet$. Given that $\Gamma[]$ is a correct positive context as well, this allows us to conclude $A \bullet C \vdash \Gamma[(B \bullet \Delta[D])]$. Finally, the residuation rule $Res_{\bullet\backslash}$ allows us to conclude $C \vdash A \backslash \Gamma[(B \bullet \Delta[D])]$ as required. The figure below summarizes the proof.

$$\dfrac{\dfrac{\vdots\, IH}{A \vdash B} \qquad \dfrac{C \vdash D}{C \vdash \Delta[D]}\,\Delta[]\ correct}{\dfrac{A \bullet C \vdash B \bullet \Delta[D]}{\dfrac{A \bullet C \vdash \Gamma[(B \bullet \Delta[D])]}{C \vdash A \backslash \Gamma[(B \bullet \Delta[D])]}\,Res_{\bullet\backslash}}\,\Gamma[]\ correct}\,Mon_{\bullet}$$

- The case for $/ \bullet P$ is symmetric.
- In case $\vdash_P \Theta[]$ is obtained by the $/ \backslash P$ rule, $\Theta[]$ is of the form $A / \Gamma[(\Delta[] \backslash B)]$ and we need to show that $\Theta[]$ is a correct positive context, that is, if $C \vdash D$, then $C \vdash A / \Gamma[(\Delta[D] \backslash B)]$ given that $\Gamma[]$ is a correct negative context, $\Delta[]$ is a correct positive context and $B \vdash A$, all by induction hypothesis.

  If $C \vdash D$ and if $\Delta[]$ is a valid positive context, then $C \vdash \Delta[D]$, which together with $B \vdash A$ and $mon_{\backslash}$ gives $\Delta[D] \backslash B \vdash C \backslash A$. This, together with the fact that $\Gamma[]$ is a correct negative context, allows us to conclude $\Gamma[(\Delta[D] \backslash B)] \vdash C \backslash A$. From this, using $Res_{\backslash\bullet}$ we can conclude $C \bullet \Gamma[(\Delta[D] \backslash B)] \vdash A$ and finally, using $Res_{\bullet/}$, $C \vdash A / \Gamma[(\Delta[D] \backslash B)]$ as required.

$$\dfrac{\dfrac{C \vdash D}{C \vdash \Delta[D]}\,\Delta[]\ correct \qquad \dfrac{\vdots\, IH}{B \vdash A}}{\dfrac{\Delta[D] \backslash B \vdash C \backslash A}{\dfrac{\Gamma[(\Delta[D] \backslash B)] \vdash C \backslash A}{\dfrac{C \bullet \Gamma[(\Delta[D] \backslash B)] \vdash A}{C \vdash A / \Gamma[(\Delta[D] \backslash B)]}\,Res_{\bullet/}}\,Res_{\backslash\bullet}}\,\Gamma[]\ correct}\,mon_{\backslash}$$

- The case for $\backslash / P$ is symmetric.

$\Longleftarrow$

Assume we have a ResMon derivation of $A \vdash B$, we show by induction on the length of the proof that there is a corresponding normal SC derivation.

We proceed by a case analysis on the last rule of the proof. The only cases which need some work are the four *Res* rules.

($Res_{\backslash\bullet}$) Suppose the last rule in the ResMon derivation is the $Res_{\backslash\bullet}$ rule.

$$\frac{B \vdash A \backslash C}{A \bullet B \vdash C}\,(Res_{\backslash\bullet})$$

By induction hypothesis, we know there is a normal SC derivation δ of $B \vdash A \backslash C$. Looking at the form of the rules, only the three following rules can have produced the sequent $B \vdash A \backslash C$: $mon\backslash$, $cont_P$ or $cont_N$. For the two *cont* rules, a further case analysis is necessary.

($mon\backslash$) If the last rule of the normal SC derivation was the monotonicity rule for $\backslash$, then we are in the following situation

$$\dfrac{\dfrac{\vdots\ \delta_1}{A \vdash B} \quad \dfrac{\vdots\ \delta_2}{D \vdash C}}{B \backslash D \vdash A \backslash C}\ mon\backslash$$

We can combine proofs δ_1 and δ_2 to produce a proof of $A \bullet (B \backslash D) \vdash C$ as follows.

$$\dfrac{\dfrac{\vdots\ \delta_2}{D \vdash C} \quad \dfrac{\dfrac{\vdots\ \delta_1}{A \vdash B} \quad \dfrac{}{\vdash_N [\,]}\,[\,]N \quad \dfrac{}{\vdash_N [\,]}\,[\,]N}{\vdash_N A \bullet (B \backslash [\,])}}{A \bullet (B \backslash D) \vdash C}\ cont_N$$

Though this new proof is not necessarily normal (the last rule of δ_2 may be $cont_N$), we can transform it into a normal proof by Lemma 4.29, which completes this case.

($cont_N$) If the last rule of the SC derivation is $cont_N$, then we are schematically in the following situation.

$$\dfrac{\dfrac{\vdots\ \delta_1}{B \vdash A \backslash C} \quad \dfrac{\vdots\ \delta_2}{\vdash_N \Gamma[]}}{\Gamma[B] \vdash A \backslash C}\ cont_N$$

We do a further case analysis on the last rule of δ_1: it can be either $mon\backslash$ or $cont_P$ ($cont_N$ and *axiom* are excluded because the proof is normal).

- If the last rule is $cont_P$, the proof is of the following form

$$\dfrac{\dfrac{\dfrac{\vdots\ \delta_1'}{A \vdash B} \quad \dfrac{\vdots\ \delta_1''}{\vdash_P \Delta[]}}{A \vdash \Delta[B]}\ cont_P \quad \dfrac{\vdots\ \delta_2}{\vdash_N \Gamma[]}}{\Gamma[A] \vdash \Delta[B]}\ cont_N$$

and we simply swap the $cont_N$ and $cont_P$ rules as follows,

$$\dfrac{\dfrac{\dfrac{\vdots\ \delta_1'}{A \vdash B} \quad \dfrac{\vdots\ \delta_2}{\vdash_N \Gamma[]}}{\Gamma[A] \vdash B}\ cont_N \quad \dfrac{\vdots\ \delta_1''}{\vdash_P \Delta[]}}{\Gamma[A] \vdash \Delta[B]}\ cont_P$$

normalize the resulting derivation if necessary and apply the appropriate case for $cont_P$ below.

- If the last rule of δ_1 is $mon\backslash$, then we are in the following situation

$$\dfrac{\dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_1'}{A \vdash B} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_1''}{D \vdash C}}{B \backslash D \vdash A \backslash C}\,mon\backslash \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_2}{\vdash_N \Gamma[]}}{\Gamma[B \backslash D] \vdash A \backslash C}\,cont_N$$

  and need to prove $A \bullet \Gamma[B \backslash D] \vdash C$, which we can do as follows.

$$\dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_1''}{D \vdash C} \qquad \dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_1'}{A \vdash B} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_2}{\vdash_N \Gamma[]} \qquad \dfrac{}{\vdash_N [\,]}\,[\,]N}{\vdash_N A \bullet \Gamma[(B \backslash [\,])]}\,\bullet\backslash N}{A \bullet \Gamma[B \backslash D] \vdash C}\,cont_N$$

$(cont_P)$ If the last rule of the SC derivation is $cont_P$, then since the positive context $\Gamma[]$ will have $\backslash$ as its main connective, derivation of the positive context $\Gamma[]$ will have as its last rule either $\backslash \bullet P$ or $\backslash / P$. We will consider each case in turn.

- In the first case, we are in are in the following situation

$$\dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_1}{B \vdash C} \qquad \dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_2}{A \vdash D} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_3}{\vdash_P \Gamma[]} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_4}{\vdash_P \Delta[]}}{\vdash_P A \backslash \Gamma[(D \bullet \Delta[])]}\,\backslash\bullet P}{B \vdash A \backslash \Gamma[(D \bullet \Delta[C])]}\,cont_P$$

  and we need to prove $A \bullet B \vdash \Gamma[(D \bullet \Delta[C])]$, which we can do as follows.

$$\dfrac{\dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_2}{A \vdash D} \qquad \dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_1}{B \vdash C} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_4}{\vdash_P \Delta[]}}{B \vdash \Delta[C]}\,cont_P}{A \bullet B \vdash D \bullet \Delta[C]}\,mon\bullet \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_3}{\vdash_P \Gamma[]}}{A \bullet B \vdash \Gamma[(D \bullet \Delta[C])]}\,cont_P$$

- In the second case, we have $cont_P$ followed by $\backslash / P$ and are in the following situation.

$$\dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_1}{B \vdash A} \qquad \dfrac{\genfrac{}{}{0pt}{}{\vdots\;\delta_2}{D \vdash C} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_3}{\vdash_N \Gamma[]} \qquad \genfrac{}{}{0pt}{}{\vdots\;\delta_4}{\vdash_P \Delta[]}}{\vdash_P \Gamma[(D / \Delta[])] \backslash C}\,\backslash / P}{B \vdash \Gamma[(D / \Delta[A])] \backslash C}\,cont_P$$

We need to give a proof of $\Gamma[(D\,/\,\Delta[A])]\bullet B\vdash C$, which we can do as follows.

$$\dfrac{\begin{matrix}\vdots\,\delta_2\\ D\vdash C\end{matrix}\qquad \dfrac{\dfrac{\begin{matrix}\vdots\,\delta_1\\ B\vdash C\end{matrix}\quad\begin{matrix}\vdots\,\delta_4\\ \vdash_P \Delta[]\end{matrix}}{B\vdash\Delta[A]}\,cont_P \qquad \begin{matrix}\vdots\,\delta_3\\ \vdash_N\Gamma[]\end{matrix}\quad \dfrac{}{\vdash_N []}\,[]N}{\vdash_N \Gamma[([]/\Delta[A])\bullet B]}\,\bullet/N}{\Gamma[(D\,/\,\Delta[A])]\bullet B\vdash C}\,cont_N$$

$(Res_{/\bullet})$ Symmetric to the case for $Res_{\backslash\bullet}$

$(Res_{\bullet\backslash})$ We need to show that whenever we have a normal proof δ of $A\bullet B\vdash C$, we can transform it into a proof of $B\vdash A\backslash C$. We do a case analysis on the last rule of δ as before. The only rules which can have applied to form a sequent $A\bullet B\vdash C$ are the monotonicity rule for the product, the positive context rule and the negative context rule, with the context rules requiring a further case analysis. We treat the cases in the indicated order.

$(\bullet mon)$ In case the last rule was $\bullet mon$, we have a proof of the following form.

$$\dfrac{\begin{matrix}\vdots\,\delta_1\\ A\vdash C\end{matrix}\quad\begin{matrix}\vdots\,\delta_2\\ B\vdash D\end{matrix}}{A\bullet B\vdash C\bullet D}\,\bullet mon$$

We can combine the subproofs into a proof of $B\vdash A\backslash(C\bullet D)$ as follows.

$$\dfrac{\begin{matrix}\vdots\,\delta_2\\ B\vdash D\end{matrix}\qquad \dfrac{\begin{matrix}\vdots\,\delta_1\\ A\vdash C\end{matrix}\quad \dfrac{}{\vdash_P []}\,[]P\quad \dfrac{}{\vdash_P []}\,[]P}{\vdash_P A\backslash(C\bullet[])}\,\backslash\bullet P}{B\vdash A\backslash(C\bullet D)}\,cont_P$$

$(cont_P)$ If the last rule of the proof is $cont_P$, then we are in the following situation.

$$\dfrac{\begin{matrix}\vdots\,\delta_1\\ A\bullet B\vdash C\end{matrix}\quad\begin{matrix}\vdots\,\delta_2\\ \vdash_P\Delta[]\end{matrix}}{A\bullet B\vdash\Delta[C]}\,cont_P$$

We do a further case analysis on the last rule of the subproof δ_1, which can be either $\bullet mon$ or $cont_N$ ($cont_P$ is excluded because the proof is normal). We treat both subcases.

- If the last rule of δ_1 is $\bullet mon$, then the proof looks as follows.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\delta_1'\\ A\vdash C\end{array}\quad \begin{array}{c}\vdots\,\delta_1''\\ B\vdash D\end{array}}{A\bullet B\vdash C\bullet D}\,\bullet mon \qquad \begin{array}{c}\vdots\,\delta_2\\ \vdash_P \Delta[]\end{array}}{A\bullet B\vdash \Delta[C\bullet D]}\,cont_P$$

We need to combine the subproofs to create a proof of $B\vdash A\backslash\Gamma[C\bullet D]$, which we can do as follows.

$$\dfrac{\begin{array}{c}\vdots\,\delta_1''\\ B\vdash D\end{array}\qquad \dfrac{\begin{array}{c}\vdots\,\delta_1'\\ A\vdash C\end{array}\quad \begin{array}{c}\vdots\,\delta_2\\ \vdash_P \Delta[]\end{array}\quad \dfrac{}{\vdash_P []}\,[]P}{\vdash_P A\backslash\Delta[(C\bullet[])]}\,\backslash\bullet P}{B\vdash A\backslash\Delta[C\bullet D]}\,cont_P$$

- If the last rule of δ_1 is $cont_N$, then the proof looks as follows.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\delta_1'\\ A\vdash B\end{array}\quad \begin{array}{c}\vdots\,\delta_1''\\ \vdash_N \Gamma[]\end{array}}{\Gamma[A]\vdash B}\,cont_N \qquad \begin{array}{c}\vdots\,\delta_2\\ \vdash_P \Delta[]\end{array}}{\Gamma[A]\vdash\Delta[B]}\,cont_P$$

and we swap the $cont_P$ and $cont_N$ rules, similar to the way in which we treated the situation in the $Res\backslash\bullet$ section of the proof, as follows

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\delta_1'\\ A\vdash B\end{array}\quad \begin{array}{c}\vdots\,\delta_2\\ \vdash_P \Delta[]\end{array}}{A\vdash\Delta[B]}\,cont_P \qquad \begin{array}{c}\vdots\,\delta_1''\\ \vdash_N \Gamma[]\end{array}}{\Gamma[A]\vdash\Delta[B]}\,cont_N$$

normalize, then continue with case $cont_N$ described below.

($cont_N$) For the case $cont_N$, we know the negative context must have $\bullet$ as its outermost symbol. This means only rules $\bullet\backslash N$ and $\backslash\bullet N$ can apply.

- Suppose our proof ends with the combination $\bullet\backslash N$, $cont_N$ as shown below

$$\dfrac{\begin{array}{c}\vdots\,\delta_1\\ D\vdash C\end{array}\qquad \dfrac{\begin{array}{c}\vdots\,\delta_2\\ A\vdash B\end{array}\quad \begin{array}{c}\vdots\,\delta_3\\ \vdash_N \Gamma[]\end{array}\quad \begin{array}{c}\vdots\,\delta_4\\ \vdash_N \Delta[]\end{array}}{\vdash_N A\bullet\Gamma[(B\backslash\Delta[])]}\,\bullet\backslash N}{A\bullet\Gamma[(B\backslash\Delta[D])]\vdash C}\,cont_N$$

then we can transform it into a proof of $\Gamma[(B\backslash\Delta[D])]\vdash A\backslash C$ as follows.

$$\dfrac{\dfrac{\vdots\,\delta_2}{A \vdash B} \quad \dfrac{\dfrac{\vdots\,\delta_1}{D \vdash C} \quad \dfrac{\vdots\,\delta_4}{\vdash_N \Delta[]}}{\Delta[D] \vdash C}\,cont_N}{B \backslash \Delta[D] \vdash A \backslash C}\,Mon\backslash \quad \dfrac{\vdots\,\delta_3}{\vdash_N \Gamma[]}$$
$$\overline{\Gamma[(B \backslash \Delta[D])] \vdash A \backslash C}\;cont_N$$

- Finally, suppose our proof ends with the combination $\bullet / N$, $cont_N$. This means the proof ends as follows.

$$\dfrac{\dfrac{\vdots\,\delta_1}{A \vdash C} \quad \dfrac{\dfrac{\vdots\,\delta_2}{B \vdash D} \quad \dfrac{\vdots\,\delta_3}{\vdash_N \Gamma[]} \quad \dfrac{\vdots\,\delta_4}{\vdash_N \Delta[]}}{\vdash_N \Gamma[(\Delta[] / D)] \bullet B}\,\bullet / N}{\Gamma[(\Delta[A] / D)] \bullet B \vdash C}\,cont_N$$

We transform it, as required, into a proof of $B \vdash \Gamma[(\Delta[A] / D)] \backslash C$ as follows.

$$\dfrac{\dfrac{\vdots\,\delta_2}{B \vdash D} \quad \dfrac{\dfrac{\dfrac{\vdots\,\delta_1}{A \vdash C} \quad \dfrac{\vdots\,\delta_4}{\vdash_N \Delta}}{\Delta[A] \vdash C}\,cont_N \quad \dfrac{\vdots\,\delta_3}{\vdash_N \Gamma[]} \quad \dfrac{}{\vdash_P []}\,[]P}{\vdash_P \Gamma[(\Delta[A] / [])] \backslash C}\,\backslash / P}{B \vdash \Gamma[(\Delta[A] / D)] \backslash C}\,cont_P$$

($Res_{\bullet /}$) Symmetric to the case for $Res_{\bullet \backslash}$ □

4.6.3 A Theorem Proving Algorithm

Now that we have established the equivalence of the context calculus SC with the combinator presentation of NL, we will take some time to give a very rough upper bound on the complexity of proof search for SC.

We first define the size of formulae, contexts and sequents. This corresponds simply to the number of symbols other than '(' and ')' we use to write it down. The context marker '[]' is counted as a single symbol.

Definition 4.31. *Let F be a formula or a context, the* size *of F is defined as follows.*

$$\begin{aligned} size([]) &= 1 \\ size(p) &= 1 \\ size(A \bullet B) &= size(A) + size(B) + 1 \\ size(A \backslash B) &= size(A) + size(B) + 1 \\ size(A / B) &= size(A) + size(B) + 1 \end{aligned}$$

The size *of a sequent $A \vdash B$ is defined as $size(A) + size(B)$.*

By inspection of the rules, it is clear that in any SC proof of a sequent $A \vdash B$, the formulae and the contexts which can appear in the proof are all subformulae or subcontexts of the sequent $A \vdash B$.

Now for a given sequent $A \vdash B$ of size n, the number of subformulae is bounded by n and the number of subcontexts is bounded by n^2: if we write the sequent as a tree, this tree will have n nodes. Apart from the root node, which corresponds to the sequent symbol, each node in the tree corresponds to a subformula and each pair of nodes such that the second node is a descendant of this first corresponds to a context.

So a proof search algorithm will require $O(n^2)$ space to store all contexts and pairs of formulae which can appear in an SC proof. The following is a very naive tabular search algorithm which decides whether or not a sequent $A \vdash B$ is derivable. It is not difficult to make the algorithm a bit smarter, for example by taking into account the polarities of formulae and contexts and by storing only sequents and contexts which have an equal number of positive and negative occurrences of all atomic formulae.

1. Given a sequent $A \vdash B$, store all pairs of subformulae and all subcontexts of $A \vdash B$.
2. Mark all instantiations of the axioms and all empty contexts as derivable.
3. For each rule which has all its premises marked as derivable but which does not have its conclusion sequent/context marked as derivable, mark its conclusion sequent or its conclusion context as derivable.
4. Repeat step 3 until no further derivable sequents/contexts are added.
5. Answer "yes" if the sequent $A \vdash B$ is marked as derivable and "no" otherwise.

For each rule, except the $cont_N$ and $cont_P$ rules, the size of the conclusion is equal to the size of its premises plus two. This gives us a maximum of $\frac{1}{2}n$ rule applications other than *cont* to produce a sequent of size n. In addition, in a normal proof there are never two consecutive $cont_N$ rules and never two consecutive $cont_P$ rules, bounding the number of *cont* rules by two thirds of the total number of rules, giving us a maximum of $\frac{3}{2}n$ iterations of step 3 of the algorithm. Now there are at most $O(n^2)$ sequents and contexts which are marked as derivable and 11 rules (all rules except the three axioms) for which we have to verify if:

1. this context is one of its premises,
2. the other premises are marked as derivable,
3. the conclusion is in the search space but not yet marked as derivable.

All of the above steps take $O(1)$. So we can conclude that it takes $O(\frac{3}{2}n * 11 * n^2) = O(n^3)$ steps to decide whether or not a given sequent is derivable using the context calculus.

4.6.4 NL without Product

Aarts and Trautwein (1995) have given an earlier proof of polynomial *parsing* for product-free NL. Figure 4.11 shows the calculus proposed by Aarts and Trautwein.

$$\frac{\Gamma,B,\Gamma' \vdash D \quad C \vdash A}{\Gamma,C,A\backslash B,\Gamma' \vdash D}\backslash'_h \qquad \frac{\Gamma,B,\Gamma' \vdash D \quad C \vdash A}{\Gamma,B/A,C,\Gamma' \vdash D}/'_h$$

$$\frac{A \vdash B \quad C \vdash D}{A \vdash (D/B)\backslash C}\backslash lift \qquad \frac{A \vdash B \quad C \vdash D}{A \vdash D/(B\backslash C)}/lift$$

$$\frac{A \vdash B \quad C \vdash D}{B\backslash C \vdash A\backslash D}\backslash mon \qquad \frac{A \vdash B \quad C \vdash D}{A/D \vdash B/C}/mon$$

$$\frac{}{A \vdash A}\,axiom$$

Fig. 4.11. Calculus for product-free NL (from Aarts and Trautwein (1995))

A remarkable feature of this calculus is that, as is clear from the formulation of the $\backslash'_h$ and $/'_h$ rules, it works using *lists* of formulae: it does not require the bracketing of the antecedent term as its input and, given a parse in the calculus of Aarts and Trautwein, we can easily extract the antecedent term simply by adding parentheses around the two formulae for each application of an $\backslash'_h$ and $/'_h$ rule.

From the perspective of the context calculus we have seen in the previous section, the product-free calculus corresponds to the following restriction on formulae

$$C \quad ::= \quad (C \bullet C) \;\Big|\; F$$

$$F \quad ::= \quad \mathsf{P} \;\Big|\; F\backslash F \;\Big|\; F/F$$

with sequents being of the form $C \vdash F$. C is simply an antecedent term written using product formulae instead of parentheses and comma's in order to make the comparison with SC more evident. So the resulting calculus still has product formulae, but only on the outside (not inside the scope of one of the implications).

Keeping this formula restriction in mind when looking at the rules for the calculus SC (Figure 4.10 on page 131), we notice the following:

- the $\bullet mon$ rule can never apply, since we cannot have a product formula as the succedent,
- neither the $\backslash \bullet P$ nor the $/ \bullet P$ rule can apply, since they necessarily have a product formula as a subformula of an implication
- for the $\bullet \backslash N$ and the $\bullet / N$ rules, the context $\Delta[]$ must be empty, since a negative context always has a product formula as its main connective and $\Delta[]$ occurs as a subformula of an implication

Sequent Rules

$$\frac{}{A \vdash A}\ axiom$$

$$\frac{A \vdash B \quad C \vdash D}{B \backslash C \vdash A \backslash D}\ \backslash mon \qquad \frac{A \vdash B \quad C \vdash D}{A / D \vdash B / C}\ / mon$$

$$\frac{A \vdash B \quad \vdash_N \Gamma[\,]}{\Gamma[A] \vdash B}\ cont_N \qquad \frac{A \vdash B \quad \vdash_P \Gamma[\,]}{A \vdash \Gamma[B]}\ cont_P$$

Negative Context Rules

$$\frac{}{\vdash_N [\,]}\ [\,]N$$

$$\frac{A \vdash B \quad \vdash_N \Gamma[\,]}{\vdash_N (A \bullet \Gamma[(B \backslash [\,])])}\ \bullet \backslash N \qquad \frac{A \vdash B \quad \vdash_N \Gamma[\,]}{\vdash_N (\Gamma[([\,] / B)] \bullet A)}\ \bullet / N$$

Positive Context Rules

$$\frac{}{\vdash_P [\,]}\ [\,]P$$

$$\frac{B \vdash A \quad \vdash_P \Delta[\,]}{\vdash_P (A / (\Delta[\,] \backslash B))}\ / \backslash P \qquad \frac{B \vdash A \quad \vdash_N \vdash_P \Delta[\,]}{\vdash_P ((B / \Delta[\,]) \backslash A)}\ \backslash / P$$

Fig. 4.12. The context calculus SC from Figure 4.10 on page 131 with product formulae occurring only on the outside

- for the $\backslash / P$ and $/ \backslash P$ rules, the negative context $\Gamma[]$ must be empty by the same reasoning.

If we take all of these restrictions into account, the reduced calculus SC for NL without product is shown in Figure 4.12.

In order to show the equivalence of the two systems, we only need to show the following:

1. the positive context rules correspond to $\backslash lift$ and $/ lift$,
2. the negative context rules correspond to $\backslash'_h$ and $/'_h$.

Item 1 is trivial; item 2 is easy to see once we realize that the two calculi work in the opposite direction: in de Groote's SC, the argument of the implication which is reduced first is always one of the outermost formulae A of the sequent, whereas in

Aarts and Trautwein's calculus the argument which is reduced first is always of of the innermost arguments C.

From the Aarts and Trautwein calculus, it is rather easy to see that (product-free) NL generates only context-free languages. Using a strategy similar to the one used for parsing AB grammars in Section 1.4, we can generate a context-free grammar which has all words in the lexicon as terminal symbols, all formulas in our grammar as non-terminal symbols and which has as its rules:

- a rule $F \to w$ if $F \in \mathrm{Lex}(w)$,
- all instances of the proof rules in Aarts and Trautwein's calculus which the formulas in the lexicon allow, eg. if $(S/(np\backslash S))\backslash S$ is a member of $\mathrm{Lex}(\text{is\_missing})$, then the following rules will be generated.

$$
\begin{aligned}
&((S/(np\backslash S))\backslash S) \to \text{is\_missing}\\
&(np\backslash S) \to ((S/(np\backslash S))\backslash S)\\
&S \to (S/(np\backslash S))\ ((S/(np\backslash S))\backslash S)\\
&S \to np\ (np\backslash S)
\end{aligned}
$$

Though the *lift* and *mon* rules mean that the resulting context-free grammar is not in Chomsky Normal Form (the second rule shown above is an example), we can apply the standard conversion to obtain a context-free grammar in Chomsky Normal Form and apply the Cocke Kasami Younger algorithm to parse the resulting grammar.

Kandulski (1988) shows that NL with product generates only context-free languages as well.

4.7 Concluding Remarks

This completes our overview of NL. Compared to the Lambek calculus, it offers both advantages and disadvantages: there are some cases where associativity is undesirable, but other cases where it seems necessary. From a computational point of view, parsing NL grammars is simpler than parsing L grammars (though Pentus' result (Pentus, 1997), which we treated in detail in Section 2.11, shows that for a *fixed* L grammar, we can convert it to a context-free grammar and benefit from polynomial parsing).

The next chapter shows how we can combine NL and L into a single logic, to obtain a *multimodal* grammar. In addition to allowing us to specify lexically whether or not we want associativity to apply, multimodal grammars allow us to give an account of phenomena which do not have a satisfactory treatment in context-free grammars.

Exercises for Chapter 4

Exercise 4.1. Verify yourself that any proof attempt of the sequent $(A\,/\,B, B\,/\,C) \vdash A\,/\,C$ which starts with a rule application other than the $/_i$ rule produces a sequent which is invalid according to Proposition 2.6. In other words, show that for each of these rule applications there is a subproof where the number of positive occurrences of one of the formulae A, B and C is not equal to the number of negative occurrences. In the derivation shown in Example 4.8 on page 104, the two failing subsequents $(A, C) \vdash A$ has negative occurrence of C and no positive occurrences, whereas $B\,/\,C \vdash B$ has one positive occurrence of C and no negative occurrences.

Exercise 4.2. Which of the following sequents — all derivable in L — are derivable in NL as well?

1. $A\,/\,B \vdash (A\,/\,C)\,/\,(B\,/\,C)$
2. $A \vdash B\,/\,(A \backslash B)$
3. $(A\,/\,B) \bullet B \vdash A$
4. $A \backslash (B \backslash C) \vdash (B \bullet A) \backslash C$

Give a proof of all derivable sequents and show in case of underivability how all proof attempts fail.

Exercise 4.3. Using the lexicon in Section 4.2.2 on page 105, derive sentences 4.1 and 4.2 in NL and sentence 4.3 in L.

Exercise 4.4. In Section 2.5, we have seen that L requires us to state explicitly that none of the antecedents are empty. Take the following lexicon.

Word	Type(s)
very	$(n\,/\,n)\,/\,(n\,/\,n)$
interesting	$n\,/\,n$
book	n

Show that in NL "very interesting book" is derivable as an expression of type n, but "very book" isn't.

Exercise 4.5. The Italian lexicon we've seen in Example 2.2 — the relevant part is repeated below —

Word	Type(s)
cosa	$(S\,/\,(S\,/\,np))$
guarda	$(S\,/\,inf)$
passare	$(inf\,/\,np)$

allows us to derive "cosa guarda passare" in L. Show that this sentence is underivable in NL.

Exercise 4.6. Exercise 4.5 has shown that we cannot treat peripheral extraction in the same simple and elegant way we used for the Lambek calculus. Give a type assignment to *cosa* in the previous exercise which makes to sentence "cose guarda passare" derivable. Comment on this type assignment. Would it allow you to treat the sentences of Exercise 2.7?

Exercise 4.7. Example 4.5 "Bill gave flowers to Mary and a toy to the children" (on page 105) is not derivable in NL under the type assignment of $((np \backslash S) / pp) / np$ to "gave". Give a type assignment to "gave" and a type assignment to "and" which is an instantiation of $(X \backslash X) / X$ which allow us to derive Example 4.5 in NL.

Exercise 4.8. In Section 4.3 we have seen the following patterns of adverbs.

(4.13) Loren carefully read Neuromancer.
(4.14) Loren read Neuromancer carefully.
(4.15) Stewart completely destroyed his credibility.
(4.16) Stewart destroyed his credibility completely.

together with the lexicon repeated below.

Word	Type(s)
Loren	np
Stewart	np
Neuromancer	np
credibility	n
his	np / n
read	$(np \backslash S) / np$
destroyed	$(np \backslash S) / np$
carefully	$(np \backslash S) \backslash (np \backslash S)$
completely	$(np \backslash S) \backslash (np \backslash S)$

1. Give natural deduction proofs for sentences 4.14 and 4.16 in NL.
2. Give natural deduction proofs for sentences 4.13 and 4.15 in NLP.

Exercise 4.9. We have seen in Exercise 2.6 that the sentences "Someone loves everyone" and "Someone is_missing" both have two normal form natural deduction derivations in L. The lexicon is repeated below. How many normal form natural deduction derivations does each of these sentences have in NL?

Word	Type(s)
someone	$(S / (np \backslash S))$
loves	$((np \backslash S) / np)$
is_missing	$((S / (np \backslash S)) \backslash S)$
everyone	$((S / np) \backslash S)$

Comment on the difference.

Exercise 4.10. Associativity corresponds to a set of two separate rules. There are some cases where it suffices to have only one of the two rules. A canonical example are cases of what is often called *right node raising*. The sentence below is a typical example.

(4.17) Loren loved but Stewart hated Neuromancer.

"loved" and "hated" are both transitive verbs and "Neuromancer" is the objet of both of them. This is another case of polymorphic coordination, of which we have already seen examples in Exercise 1.4.4 and Section 4.2.1: "but", like' "and" can be assigned the formula $(X \setminus X) / X$ for several instantiations of X. We have seen several examples in Exercise 1.4.4.

1. What is the instantiation of X which is appropriate for the sentence above?
2. Which of the two associativity rules do we need to derive the sentence?

Exercise 4.11. To familiarize yourself with the axiomatic calculus, prove $A \vdash B / (A \setminus B)$. Though this statement has a trivial proof in the sequent calculus as well as in natural deduction, you'll find that it requires a bit more effort here.

Exercise 4.12. Prove the following lemma for the residuation calculus of Figure 4.6 on page 114 without using the $(Trans)$ rule.

Lemma 4.13. *If from $A \vdash C$ we can derive $B \vdash C$, then from $\Gamma[A] \vdash C$ we can derive $\Gamma[B] \vdash C$.*

Exercise 4.14. Reprove the case for $\setminus_h$ of Lemma 4.14 (on page 119) using Lemma 4.13 you proved for Exercise 4.12.

Exercise 4.15. Find all different proofs of the sequent

$$S / (np \setminus S), (S / (np \setminus S)) \setminus S \vdash S$$

using both de Groote's context calculus SC, as shown on Figure 4.10 on page 131, and Aarts and Trautwein's calculus, shown on Figure 4.11 on page 141.

Exercise 4.16. Prove the example sequent (from de Groote, 1999)

$$a \vdash (c / ((a \bullet b) \setminus c)) / b$$

in the context calculus SC from Figure 4.10.

References

Aarts, E., Trautwein, K.: Non-associative Lambek categorial grammar in polynomial time. Mathematical Logic Quarterly 41, 476–484 (1995)

Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science, vol. 53. Cambridge University Press (2001)

Chomsky, N.: Some concepts and consequences of the theory of Government and Binding. MIT Press, Cambridge (1982)

Chomsky, N.: The minimalist program. MIT Press, Cambridge (1995)

Došen, K.: Sequent-systems and groupoid models I. Studia Logica 47(4), 353–385 (1988)

Došen, K.: Sequent-systems and groupoid models II. Studia Logica 48(1), 41–65 (1989)

Došen, K.: A brief survey of frames for the Lambek calculus. Zeitschrift für Mathematische Logic und Grundlagen der Mathematik 38, 179–187 (1992)

Emms, M.: Parsing with polymorphism. In: Proceedings of the Sixth Conference of the European Association of Computational Linguistics, pp. 120–129 (1993)

Emms, M.: An undecidability result for polymorphic Lambek calculus. In: Dekker, P., Stokhof, M. (eds.) Proceedings 10th Amsterdam Colloquium, pp. 539–549 (1995)

de Groote, P.: The Non-associative Lambek Calculus with Product in Polynomial Time. In: Murray, N.V. (ed.) TABLEAUX 1999. LNCS (LNAI), vol. 1617, pp. 128–139. Springer, Heidelberg (1999)

Joshi, A., Schabes, Y.: Tree adjoining grammars. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, ch. 2. Springer, Berlin (1997)

Kandulski, M.: The equivalence of nonassociative Lambek categorial grammars and context free grammars. Zeitschrift für Mathematische Logic und Grundlagen der Mathematik 34, 41–52 (1988)

Kanovich, M.: The complexity of horn fragments of linear logic. Ann. Pure Appl. Logic 69(2-3), 195–241 (1994)

Kurtonina, N.: Frames and labels. A modal analysis of categorial inference. PhD thesis, OTS Utrecht, ILLC Amsterdam (1995)

Kurtonina, N.: Categorial inference and modal logic. Journal of Logic, Language and Information 7(4), 399–411 (1998)

Lambek, J.: On the calculus of syntactic types. In: Jakobson, R. (ed.) Structure of Language and its Mathematical Aspects, pp. 166–178. American Mathematical Society (1961)

Lambek, J.: Categorial and categorical grammars. In: Oehrle, R.T., Bach, E., Wheeler, D. (eds.) Categorial Grammars and Natural Language Structures. Reidel, Dordrecht (1988)

Moortgat, M., Oehrle, R.T.: Proof nets for the grammatical base logic. In: Abrusci, V.M., Casadio, C., Sandri, G. (eds.) Dynamic Perspectives in Logic and Linguistics. Cooperativa Libraria Universitaria Editrice Bologna (1999)

Pentus, M.: Product-free Lambek calculus and context-free grammars. Journal of Symbolic Logic 62(2), 648–660 (1997)

Pentus, M.: Lambek calculus is NP-complete. Theoretical Computer Science 357(1), 186–201 (2006)

Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. Center for the Study of Language and Information, Stanford (1994) (distributed by Cambridge University Press)

Savateev, Y.: Product-Free Lambek Calculus Is NP-Complete. In: Artemov, S., Nerode, A. (eds.) LFCS 2009. LNCS, vol. 5407, pp. 380–394. Springer, Heidelberg (2008)

Stabler, E.P.: Derivational Minimalism. In: Retoré, C. (ed.) LACL 1996. LNCS (LNAI), vol. 1328, pp. 68–95. Springer, Heidelberg (1997)